UNIVERSIDAD POLITÉCNICA DE CARTAGENA

Grado en Arquitectura Naval e Ingeniería de Sistemas Marinos

Asignatura: Matemáticas I

LIBRO DE PRÁCTICAS de *Maxima*

Cinco prácticas de Cálculo y Álgebra Lineal con CAS

AUTOR

David Conesa Pagán

Matrícula de Honor en Matemáticas I

UPCT ▪ Cartagena

Curso académico
2025 / 2026

"Las matemáticas son el alfabeto con el cual Dios ha escrito el universo."

— Galileo Galilei

Libro de Prácticas de Maxima

Asignatura *Matemáticas I*

Grado en Arquitectura Naval e Ingeniería de Sistemas Marinos

Universidad Politécnica de Cartagena

© David Conesa Pagán, 2026

Todos los derechos reservados sobre los contenidos originales.

Aviso legal y licencia

Esta obra está sujeta a la licencia internacional **Creative Commons Reconocimiento — NoComercial — SinObraDerivada 4.0** (**CC BY-NC-ND 4.0**). En virtud de sus términos **no se autoriza**:

- La explotación o uso *comercial* de esta obra (venta, alquiler o distribución onerosa).
- La elaboración de *obras derivadas*, traducciones, adaptaciones o transformaciones de su contenido.
- Su distribución sin la correspondiente atribución expresa al autor original.

Sí se autoriza el *compartido íntegro* (sin modificación), con fines estrictamente personales o académicos no comerciales, siempre que se reconozca la autoría. Cualquier uso no contemplado por la licencia requiere consentimiento expreso por escrito del autor.

Escanea para leer el aviso legal y la licencia íntegra.

creativecommons.org/licenses/by-nc-nd/4.0

Compilado el 8 de mayo de 2026 con LaTeX y *Maxima* 5.49.
Todas las salidas y figuras han sido generadas automáticamente desde los archivos `.wxm` originales.

Matrícula de Honor en Matemáticas I, curso 2025/2026.

Prefacio

Este libro recoge, en un único volumen ordenado y revisado, las cinco prácticas de laboratorio realizadas durante el curso de **Matemáticas I** del Grado en Arquitectura Naval e Ingeniería de Sistemas Marinos en la **Universidad Politécnica de Cartagena**. Cada práctica explora un bloque fundamental de la asignatura mediante el sistema de cálculo simbólico *Maxima*: desde las primeras operaciones aritméticas y el dibujo de funciones, pasando por la programación de algoritmos numéricos, hasta el álgebra matricial avanzada con factorizaciones LU y QR.

La obra ha sido elaborada por **David Conesa Pagán**, alumno que obtuvo la calificación de *Matrícula de Honor* en la asignatura.

Cómo leer este libro. Cada celda mantiene el formato clásico de *wxMaxima*: a la izquierda, en rojo, la etiqueta ($\%\,\mathrm{i}_n$) indicando el número de entrada, seguida del código tal cual se introduce en la terminal del programa. Inmediatamente debajo aparece el resultado, renderizado como expresión matemática, junto con su etiqueta ($\%\,\mathrm{o}_n$); las gráficas se etiquetan como ($\%\,\mathrm{t}_n$). Los enunciados destacados se presentan en cuadros (✒) y las notas (🛈) contienen aclaraciones de sintaxis o interpretación.

David Conesa Pagán *Cartagena*

Índice General

Prefacio **3**

1. Práctica 1: Primeros pasos con Maxima **1**
1.1. Operaciones aritméticas elementales . . . 1
1.2. Variables y funciones . . . 3
1.3. Gráficas de funciones. . . . 4
1.4. Límites y derivadas . . . 14
1.5. Cálculo de puntos críticos . . . 15

2. Práctica 2: Optimización de funciones **17**
2.1. Calcula los extremos absolutos de f(x)=x^3-x^2-10x+1 en [-2,4] . . . 17
2.2. Estudia la función f(x)=x^5-3x^2-x en [-1,2] . . . 18
2.3. Estudia la función f(x)=(x^2-2x) e^(-x^2/5) en [-2,2]. . . . 18
2.4. Calcula el mínimo absoluto de f(x)= x^2-8sqrt(x)+5. Repite el ejercicio cambiando la raíz cuadrada por la raíz cúbica. . . . 19
2.5. Un hilo de 10 metros se divide en dos trozos. Con uno de ellos se forma un círculo y con el otro un cuadrado. Halla la forma de hacerlo para que la suma de las áreas de ambas figuras sea mínima. . . . 21
2.6. Define la función a trozos f(x)=x^2 si x<0, x^3 si x>0 y dibuja su gráfica. . . . 22
2.7. Calcula los extremos absolutos de f(x)= 2x^3-3x-1 si x<0 (x^2 -1)e^(-x) x>0 en el intervalo [-2,2] . . . 22
2.8. Calcula el polinomio de Taylor de f(x)=cos(x) de grado 8 centrado en x=0. Dibuja conjuntamente la gráfica de f y del polinomio de Taylor. . . . 23

3. Práctica 3: Programación, iteración y método de Newton **25**
3.1. Programación básica . . . 25
3.2. Iteración de funciones . . . 27
3.3. El método de Newton para aproximar ceros de funciones . . . 30

4. Práctica 4: Matrices, vectores y subespacios **37**
4.1. Definición de matrices y operaciones básicas. . . . 37
4.2. Ejercicios propuestos . . . 43

5. Práctica 5: Diagonalización y factorizaciones **47**
5.1. Diagonalización . . . 47
5.2. Factorización LU . . . 50
5.3. Factorización QR . . . 54

CAPÍTULO 1

Práctica 1: Primeros pasos con Maxima

Resumen de la práctica

En esta primera práctica nos familiarizamos con la sintaxis básica del CAS *Maxima*: aprendemos a realizar operaciones aritméticas, a manipular variables, a definir funciones y a representarlas gráficamente. Cerramos la práctica con un primer contacto con el cálculo diferencial: límites, derivadas y puntos críticos.

1.1. Operaciones aritméticas elementales

Comencemos mostrando cómo se realizan las operaciones aritméticas básicas con MAXIMA.

Veamos ejemplos de sumas, restas, multiplicaciones, divisiones y potencias.

```
(%i1)  1+25;
```

26 (%o1)

```
(%i2)  35+48;
```

83 (%o2)

```
(%i3)  1-25;
```

−24 (%o3)

Ya sé sumar.

Para multiplicar utilizamos el punto.

```
(%i4)  3*4;
```

12 (%o4)

Para multiplicar se escribe con "*", pero en el programa sale como "·"

```
(%i5)  14*83;
```

1162 (%o5)

A diferencia de otros programas, el producto siempre se realiza con el símbolo (punto). Así, la siguiente operación no proporciona ningún resultado.

```
(%i6)  (1+3)(1+4);
apply: found 4 where a function was expected.
```

! Error: *apply: found 4 where a function was expected.* (%o6)

Hay que teclear lo siguiente:

```
(%i7)  (1+3)*(1+4);
```

20 (%o7)

Prosigamos con la división.

```
(%i8)  3/4;
```

$\frac{3}{4}$ (%o8)

Si queremos obtener una expresión decimal utilizamos el comando float.

```
(%i9) float(3/4);
```

0.75 (%o9)

Ejercicio

Calcula una expresión decimal del número (1+3/10)(2+3*5)

```
(%i10) float((1+3/10)*(2+3*5));
```

22.1 (%o10)

Las potencias se introducen con el acento circunflejo.

```
(%i11) 2^3;
```

8 (%o11)

La raíz cuadrada también se calcula con el comando sqrt

```
(%i12) sqrt(2);
```

$\sqrt{2}$ (%o12)

```
(%i13) float(sqrt(2));
```

1.4142135623730951 (%o13)

Algunas constantes matemáticas tienen una representación simbólica, consistente en su nombre precedido por el símmbolo %.

Ejercicio

Halla la suma de la raíz cuadrada de 2 y la raíz cuarta de 7.

```
(%i14) float(sqrt(2)+7^(1/4));
```

3.040790124070881 (%o14)

```
(%i15) %e;
```

e (%o15)

```
(%i16) float(%e);
```

2.718281828459045 (%o16)

```
(%i17) float(%pi);
```

3.141592653589793 (%o17)

```
(%i18) %i;
```

i (%o18)

```
(%i19) %i^2;
```

-1 (%o19)

Ejercicio

Calcula una aproximación decimal de la raíz cúbica de e+3π.

```
(%i20) float((%e+3*%pi)^(1/3));
```

2.298490489614301 (%o20)

1.2. Variables y funciones

La asignación de variables tiene la forma nombrevariable: valorvariable

```
(%i1) a:5;
```
5 (%o1)

```
(%i2) 8*a;
```
40 (%o2)

```
(%i3) a^3;
```
125 (%o3)

Podemos cambiar el valor de la variable cuando se desee.

```
(%i4) a:25;
```
25 (%o4)

Para borrar el valor de una variable utilizamos el comando kill.

```
(%i5) kill(a);
```
done (%o5)

Para borrar todas las variables escribimos

```
(%i6) kill(all);
```
done (%o6)

Ejercicio

Asignando valores a variables, define un radio de un círculo con valor 8, define su longitud y su área. Calcula también la suma de su longitud y su área.

```
(%i7) r:8;
```
8 (%o7)

```
(%i8) long:float(2*%pi*r);
```
50.26548245743669 (%o8)

```
(%i9) area:float(%pi*r^2);
```
201.06192982974676 (%o9)

```
(%i10) suma:long+area;
```
251.32741228718345 (%o10)

Las variables pueden depender a su vez de otras variables, definiendo así funciones.

```
(%i11) f:x^2;
```
x^2 (%o11)

Para evaluar en un punto, se utiliza el comando de evaluación at, como sigue:

```
(%i12) at(f,x=3);
```
9 (%o12)

Otra forma de definir funciones consiste en utilizar la asignación :=, como se muestra:

```
(%i13) f(x):=x^2;
```

$f(x) := x^2$ (%o13)

En este caso la evaluación se efectúa de la siguiente manera.

```
(%i14) f(3);
```

9 (%o14)

Ejercicio

Define la función f(x)=(x^2+1)/(e^x-3). Calcula el producto de los valores de f en x=1, x=2 y x=pi.

```
(%i15) f:(x^2+1)/(%e^x-3);
```

$$\frac{x^2+1}{e^x-3}$$ (%o15)

```
(%i16) prod:float(at(f,x=1)*at(f,x=2)*at(f,x=%pi));
```

−4.364689397911327 (%o16)

```
(%i17) prod^5;
```

−1584.0420943792305 (%o17)

```
(%i18) f(x):=(x^2+1)/(%e^x-3);
```

$$f(x) := \frac{x^2+1}{e^x-3}$$ (%o18)

```
(%i19) float(f(1)*f(2)*f(%pi));
```

−4.364689397911327 (%o19)

1.3. Gráficas de funciones.

```
(%i1) kill(all);
```

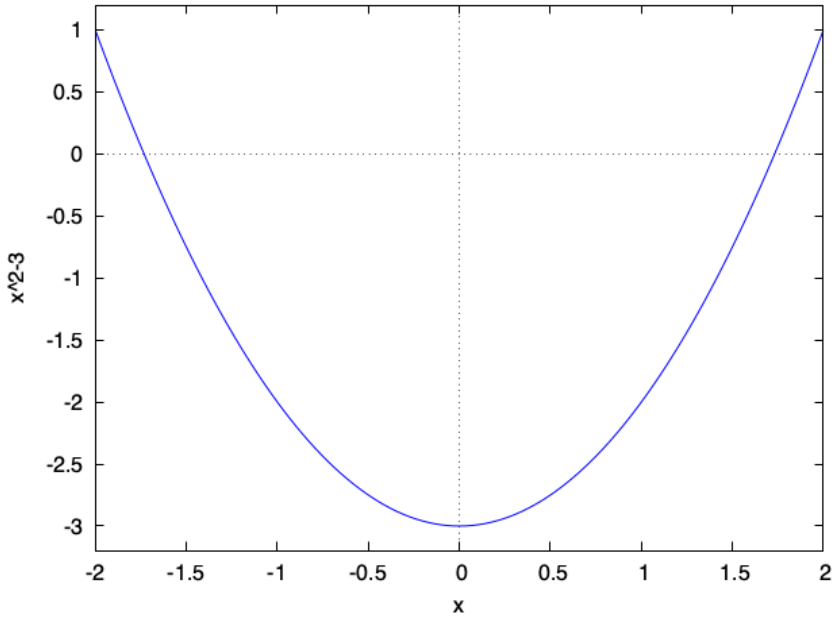
done (%o1)

Para dibujar la gráfica de la función real de una variable real utilizamos el comando wxplot2d.

```
(%i2) f:x^2-3;
```

$x^2 - 3$ (%o2)

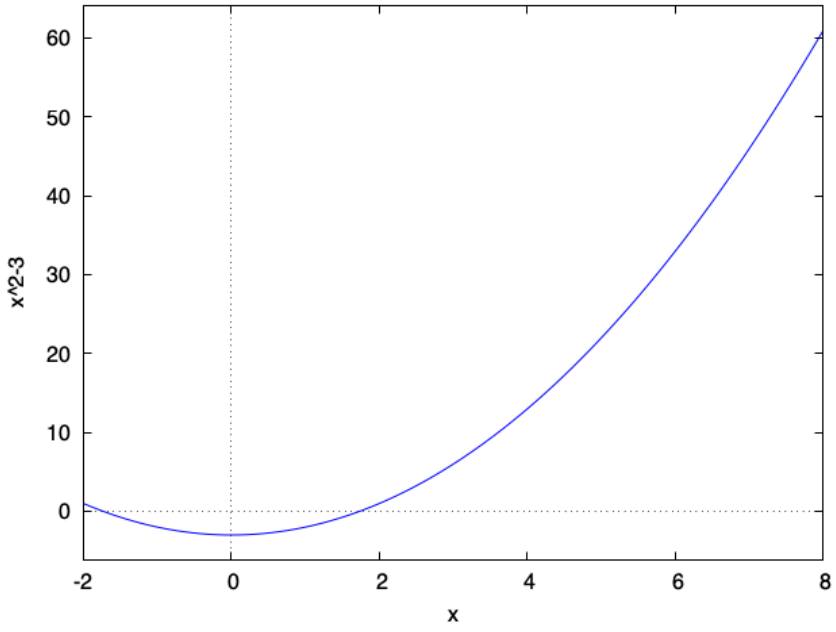
```
(%i3) wxplot2d(f, [x,-2,2]);
```



(%t3)

Podemos cambiar el rango en el que dibujamos la gráfica.

(%i4)
```
wxplot2d(f, [x,-2,8]);
```



(%t4)

Ejercicio

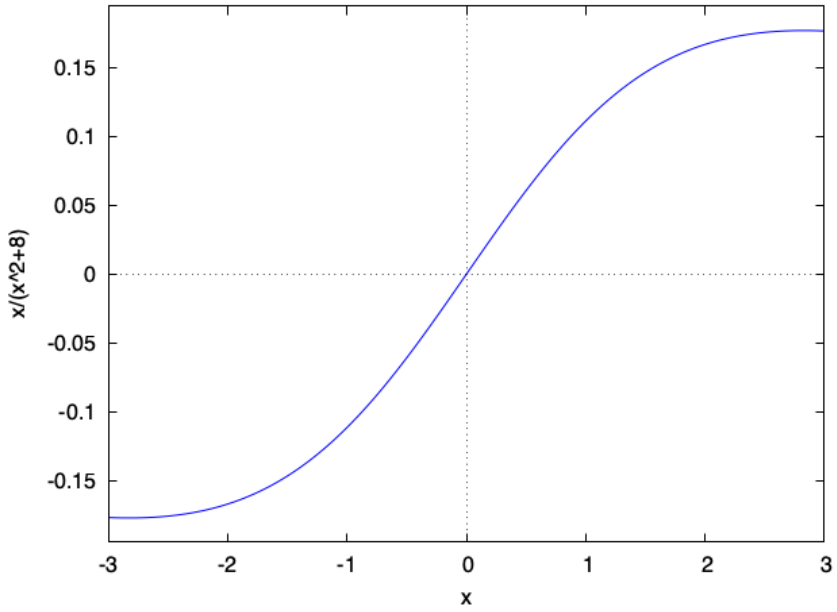
Define a función f(x)=x/(x^2+8) y dibuja su gráfica en el intervalo [-3,3]

(%i5)
```
f:x/(x^2+8);
```

$$\frac{x}{x^2+8}$$

(%o5)

(%i6)
```
wxplot2d(f,[x,-3,3]);
```



(%t6)

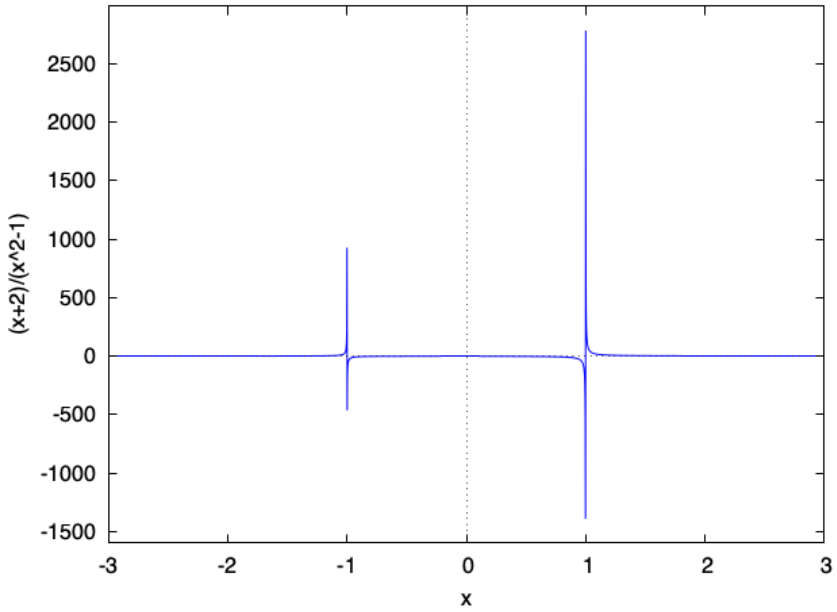
Dibujemos la gráfica de una función racional.

(%i7)
```
f:(x+2)/(x^2-1);
```

$$\frac{x+2}{x^2-1}$$

(%o7)

(%i8)
```
wxplot2d(f, [x,-3,3]);
```

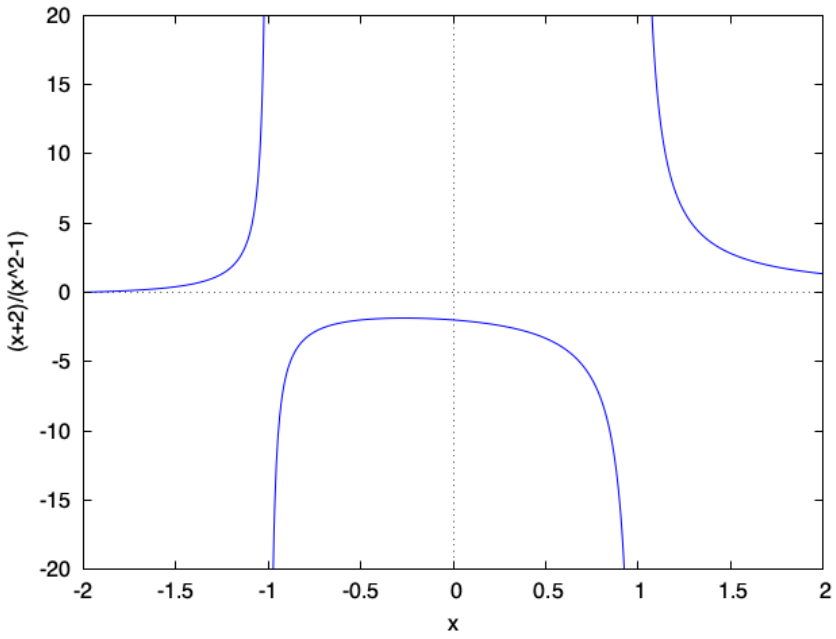


(%t8)

Acotemos el valor de y para tener una mejor representación gráfica.

(%i9)
```
wxplot2d(f,[x,-2,2],[y,-20,20]);
```

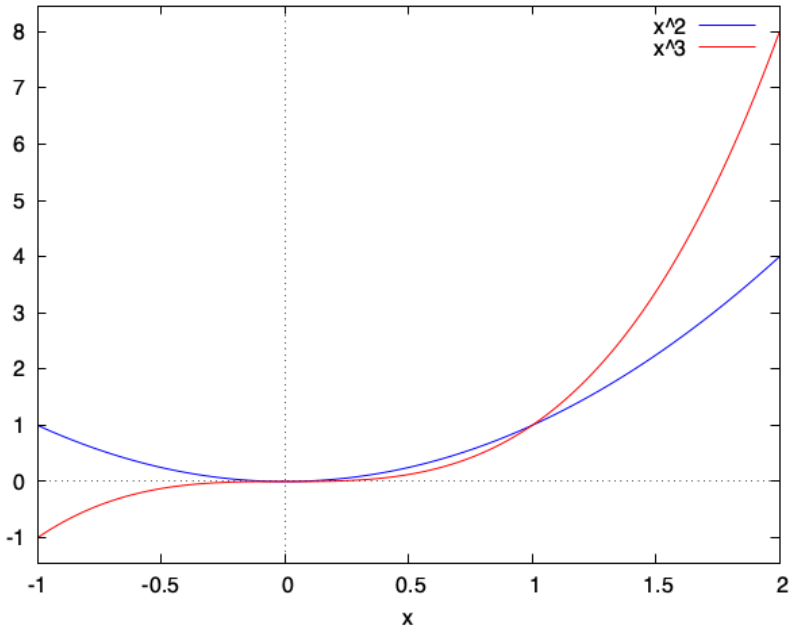
```
expt: no definido: desde 0 a un exponente negativo.  expt: no definido: desde 0
a un exponente negativo.  evaluaciones de expresiones para valor no numérico en
algún lugar en el rango del gráfico.  plot2d: some values will be clipped.
```



(%t9)

Para dibujar varias gráficas conjuntamente se introducen las funciones entre corchetes.

(%i10) `wxplot2d([x^2,x^3],[x,-1,2]);`



(%t10)

Veamos las funciones elementales y sus gráficas.

Empecemos por las funciones trigonométricas.

(%i11) `sin(%pi);`

0 (%o11)

(%i12) `sin(3);`

$\sin 3$ (%o12)

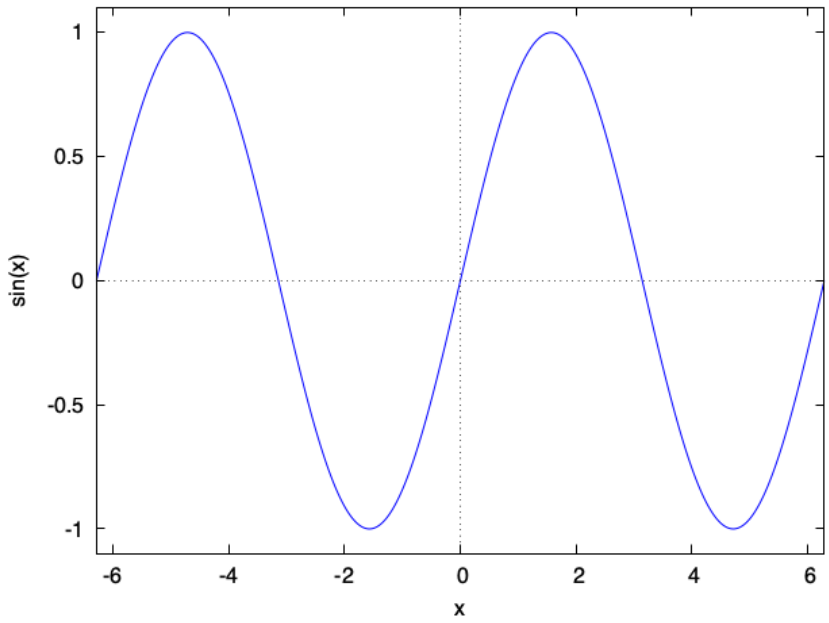
(%i13) `sin(3.0);`

0.1411200080598672 (%o13)

(%i14) `cos(%pi);`

-1 (%o14)

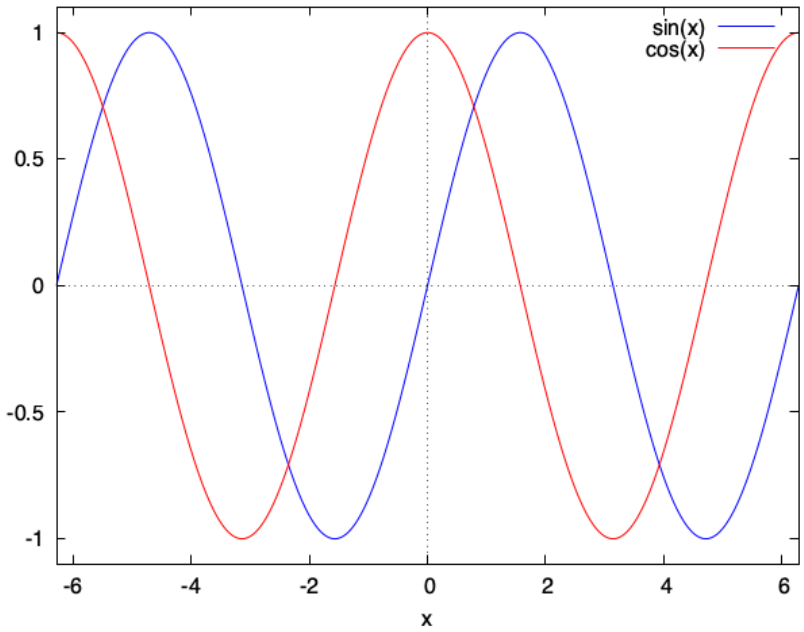
```
(%i15) wxplot2d(sin(x),[x,-2*%pi,2*%pi]);
```



(%t15)

Dibujemos la gráfica del seno y del coseno conjuntamente.

```
(%i16) wxplot2d([sin(x),cos(x)],[x,-2*%pi,2*%pi]);
```



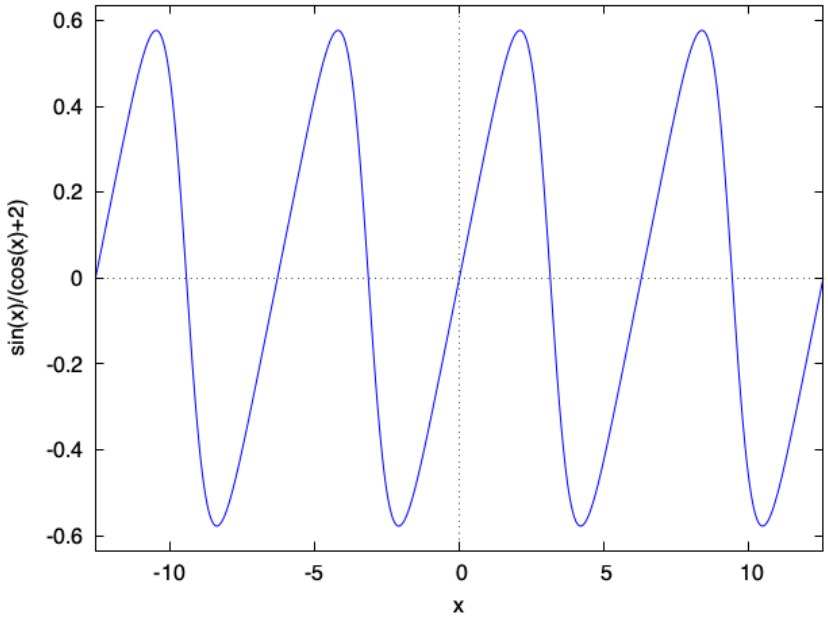
(%t16)

Define la función f(x)=sen(x)/(2+cos(x)) y dibuja su gráfica en el intervalo [-4pi,4pi].

```
(%i17) f:sin(x)/(2+cos(x));
```

$$\frac{\sin x}{\cos x + 2}$$ (%o17)

```
(%i18) wxplot2d(f,[x,-4*%pi,4*%pi]);
```



(%t18)

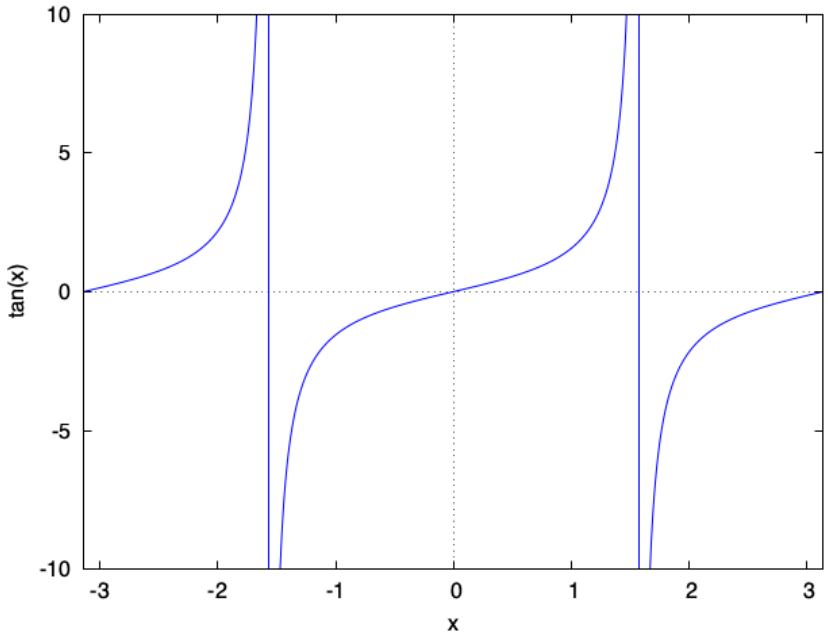
Sigamos con otras funciones trigonométricas.

(%i19) `tan(2.0);`

-2.185039863261519 (%o19)

(%i20) `wxplot2d(tan(x), [x,-%pi,%pi], [y,-10,10]);`

```
plot2d: some values will be clipped.
```

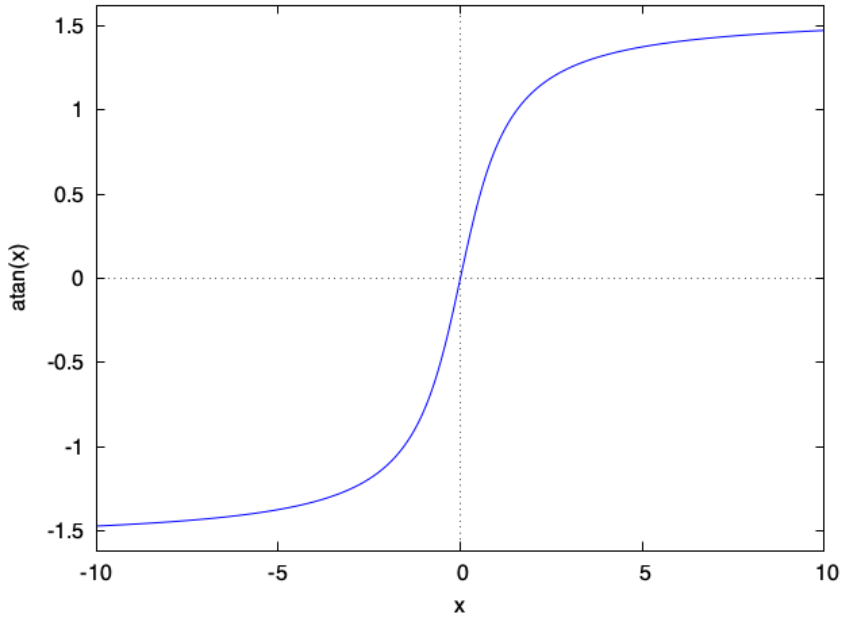


(%t20)

(%i21) `atan(1);`

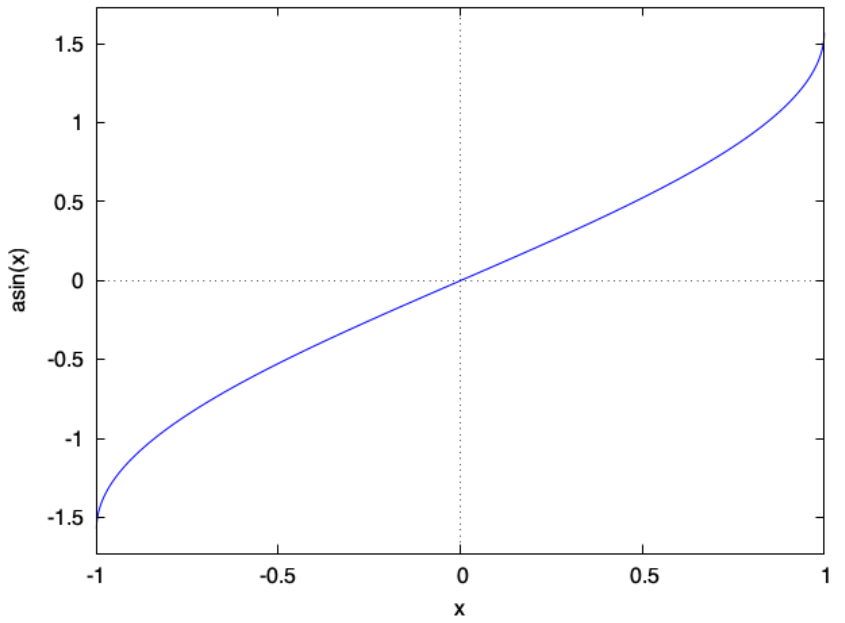
$\frac{\pi}{4}$ (%o21)

(%i22) `wxplot2d(atan(x),[x,-10,10]);`



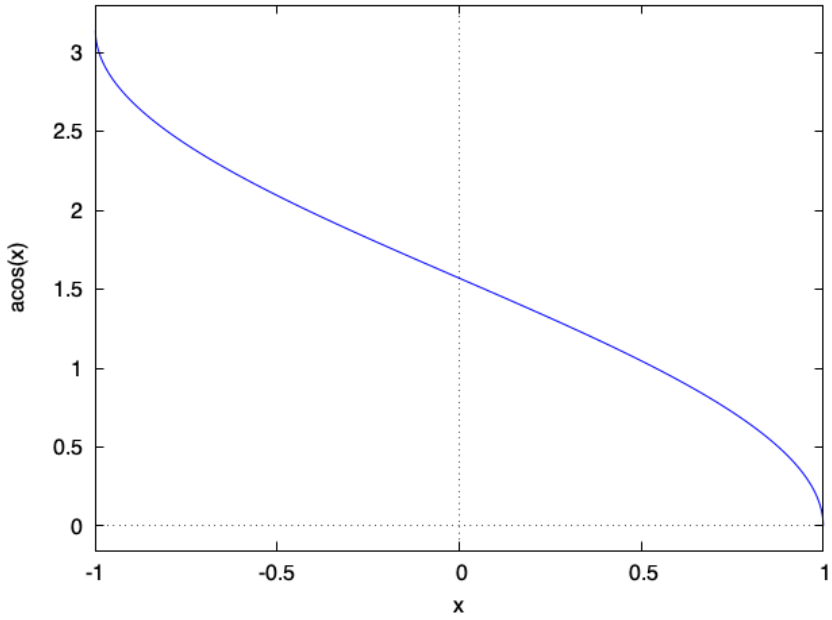
(%t22)

(%i23) `wxplot2d(asin(x),[x,-1,1]);`



(%t23)

(%i24) `wxplot2d(acos(x),[x,-1,1]);`



(%t24)

Veamos ahora las funciones exponenciales y logarítmicas.

(%i25)
```
exp(5);
```

e^5 (%o25)

(%i26)
```
exp(5.0);
```

148.4131591025766 (%o26)

(%i27)
```
%e^(5.0);
```

148.4131591025766 (%o27)

(%i28)
```
log(1);
```

0 (%o28)

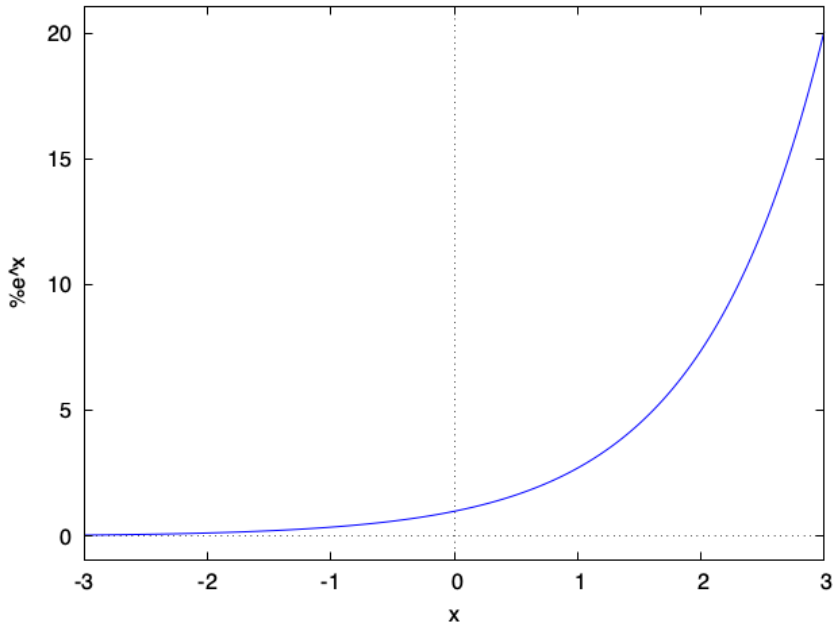
(%i29)
```
log(-1);
```

$\log(-1)$ (%o29)

(%i30)
```
log(80.0);
```

4.382026634673881 (%o30)

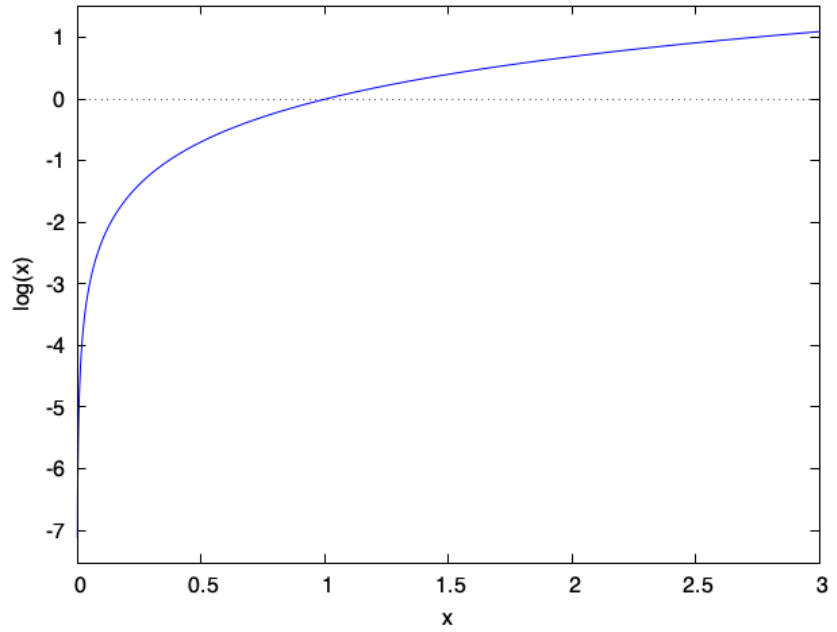
(%i31)
```
wxplot2d(exp(x),[x,-3,3]);
```



(%t31)

(%i32) `wxplot2d(log(x),[x,0,3]);`

```
encontraba un log(0).  evaluaciones de expresiones para valor no numérico en
algún lugar en el rango del gráfico.
```



(%t32)

Ejercicio

Define la función f(x)=arctg(x)/(5e^x+log(x+23)) y dibuja su gráfica en [-3,3]

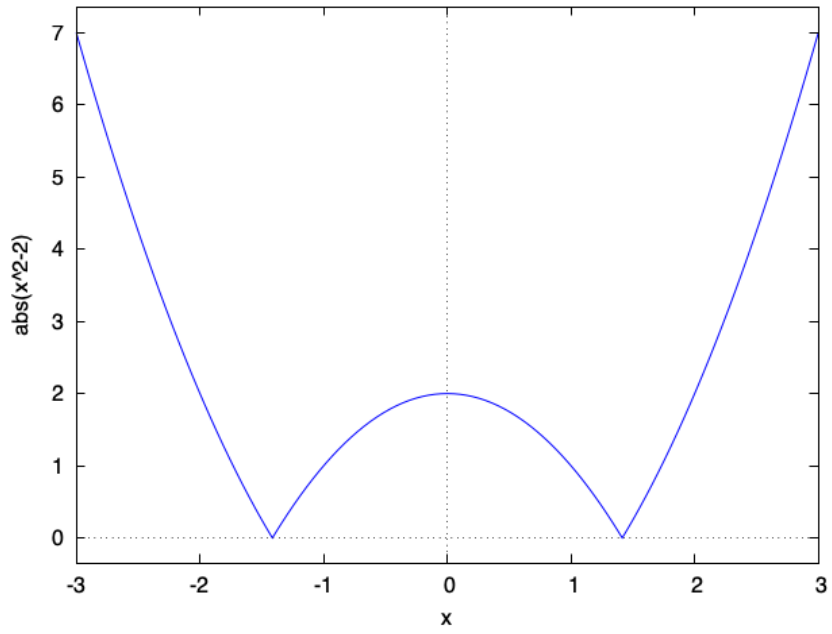
(%i33) `f:atan(x)/(5*exp(x)+log(x+23));`

$$\frac{\arctan x}{\log\left(x+23\right)+5\,e^{x}}$$

(%o33)

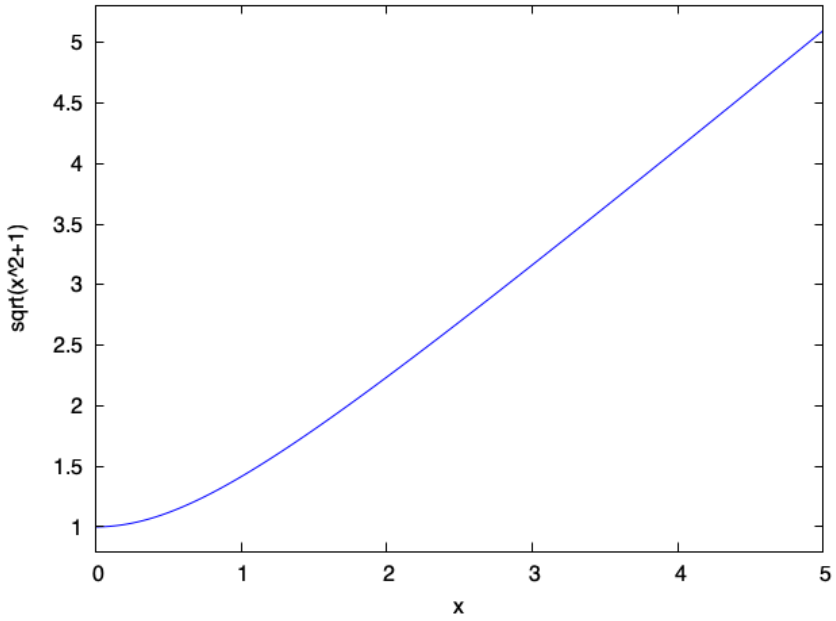
Finalmente mostremos el valor absoluto y la raíz cuadrada.

(%i34) `wxplot2d(abs(x^2-2),[x,-3,3]);`



(%t34)

(%i35) `wxplot2d(sqrt(x^2+1),[x,0,5]);`



(%t35)

Ejercicio

Define la función f(x)=e^(-x)/(2+sen(x)) y dibuja su gráfica entre -2 y 2. ¿Cuánto vale la función f en x=3/2?

(%i36) `f:exp(-x)/(2+sin(x));`

$$\frac{e^{-x}}{\sin x + 2}$$

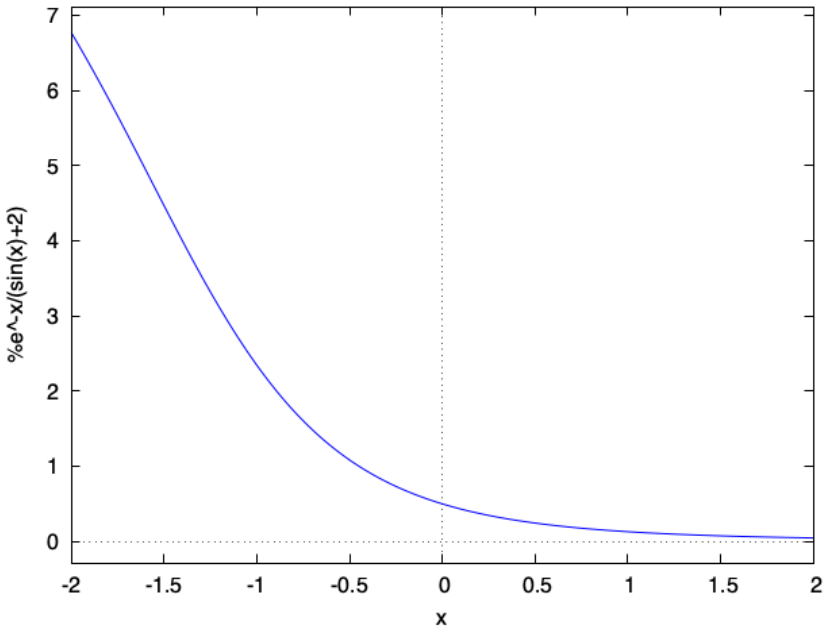
(%o36)

(%i37) `float(at(f,x=3/2));`

0.07443887684403443

(%o37)

(%i38) `wxplot2d(f,[x,-2,2]);`



(%t38)

1.4. Límites y derivadas

(%i1)
```
kill(all);
```
done (%o1)

Definamos la siguiente función:

(%i2)
```
f:(x-3)/(x^2-4);
```
$$\frac{x-3}{x^2-4}$$ (%o2)

Para calcular su derivada introducimos:

(%i3)
```
df:diff(f,x);
```
$$\frac{1}{x^2-4}-\frac{2\,(x-3)\,x}{\left(x^2-4\right)^2}$$ (%o3)

(%i4)
```
ratsimp(df);
```
$$-\left(\frac{x^2-6\,x+4}{x^4-8\,x^2+16}\right)$$ (%o4)

Su segunda derivada se calcula así:

(%i5)
```
diff(f,x,2);
```
$$-\left(\frac{4\,x}{\left(x^2-4\right)^2}\right)-\frac{2\,(x-3)}{\left(x^2-4\right)^2}+\frac{8\,(x-3)\,x^2}{\left(x^2-4\right)^3}$$ (%o5)

(%i6)
```
ratsimp(diff(f,x,2));
```
$$\frac{2\,x^3-18\,x^2+24\,x-24}{x^6-12\,x^4+48\,x^2-64}$$ (%o6)

Ejercicio

Define la función f(x)=sen(x)/(2+cos(x)) y derivalá.

(%i7)
```
f:sin(x)/(2+cos(x));
```

$$\frac{\sin x}{\cos x + 2} \qquad (\%o_7)$$

```
(%i8) df:ratsimp(diff(f,x));
```

$$\frac{\sin^2 x + \cos^2 x + 2\cos x}{\cos^2 x + 4\cos x + 4} \qquad (\%o_8)$$

Veamos algunos límites.

```
(%i9) limit(f, x, 0);
```

$$0 \qquad (\%o_9)$$

```
(%i10) limit(f,x,2);
```

$$\frac{\sin 2}{\cos 2 + 2} \qquad (\%o_{10})$$

```
(%i11) limit(f,x,inf);
```

$$ind \qquad (\%o_{11})$$

```
(%i12) limit(f,x,minf);
```

$$ind \qquad (\%o_{12})$$

```
(%i13) limit(sin(x)^3/(x^2),x,0);
```

$$0 \qquad (\%o_{13})$$

```
(%i14) limit(x/log(x)^x,x,inf);
```

$$0 \qquad (\%o_{14})$$

```
(%i15) limit(sqrt(x^2+1)-sqrt(x),x,inf);
```

$$\infty \qquad (\%o_{15})$$

Calcula el límite en x=0 de la función f(x)=(x^2+5x)/(e^x-1)

```
(%i16) limit((x^2+5*x)/(e^x-1),x,0);
```

$$\frac{5}{\log e} \qquad (\%o_{16})$$

1.5. Cálculo de puntos críticos

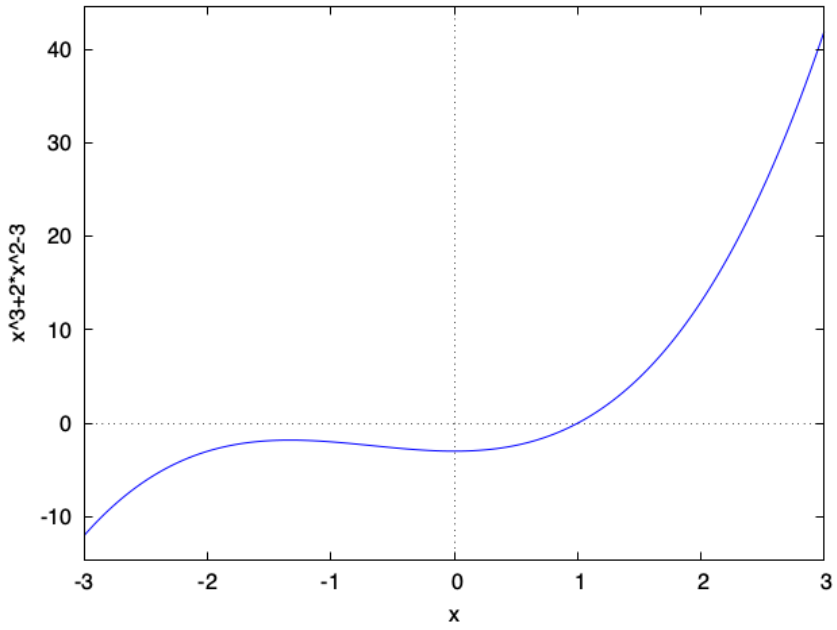
Para resolver la ecuación f'(x)=0 (o cualquier ecuación), se utiliza el comando solve. Veamos un ejemplo paso a paso.

Sea

```
(%i1) f:x^3+2*x^2-3;
```

$$x^3 + 2x^2 - 3 \qquad (\%o_1)$$

```
(%i2) wxplot2d(f,[x,-3,3]);
```



(%t2)

Calculamos su derivada y sus puntos críticos.

(%i3) `df:diff(f,x);`

$$3\,x^2+4\,x$$ (%o3)

(%i4) `s:solve(df=0,x);`

$$\left[x=-\left(\frac{4}{3}\right),x=0\right]$$ (%o4)

Evaluamos la función en los puntos críticos:

(%i5) `v1:at(f,s[1]);`

$$-\left(\frac{49}{27}\right)$$ (%o5)

(%i6) `v2:at(f,s[2]);`

$$-3$$ (%o6)

CAPÍTULO 2

Práctica 2: Optimización de funciones

Resumen de la práctica

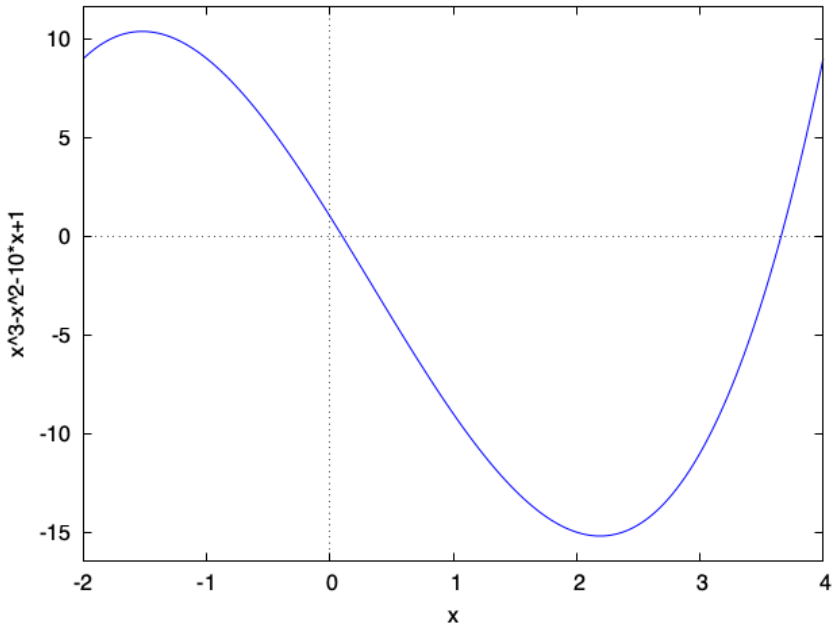
Esta segunda práctica está enteramente dedicada al **cálculo de extremos absolutos** de funciones reales de variable real, incluyendo funciones a trozos y problemas de optimización con restricciones (geometría, longitudes, áreas). Se cierra con un primer contacto con los **polinomios de Taylor**.

2.1. Calcula los extremos absolutos de f(x)=x^3-x^2-10x+1 en [-2,4]

(%i1) `f:x^3-x^2-10*x+1;`

$x^3 - x^2 - 10\,x + 1$ (%o1)

(%i2) `wxplot2d(f,[x,-2,4]);`



(%t2)

(%i3) `df:diff(f,x);`

$3\,x^2 - 2\,x - 10$ (%o3)

(%i4) `pc:float(solve(df=0,x));`

$[x = -1.5225881209433405, x = 2.189254787610007]$ (%o4)

(%i5) `min:at(f,pc[2]);`

-15.192644092424496 (%o5)

(%i6) `max:at(f,pc[1]);`

10.377829277609678 (%o6)

2.2. Estudia la función f(x)=x^5-3x^2-x en [-1,2]

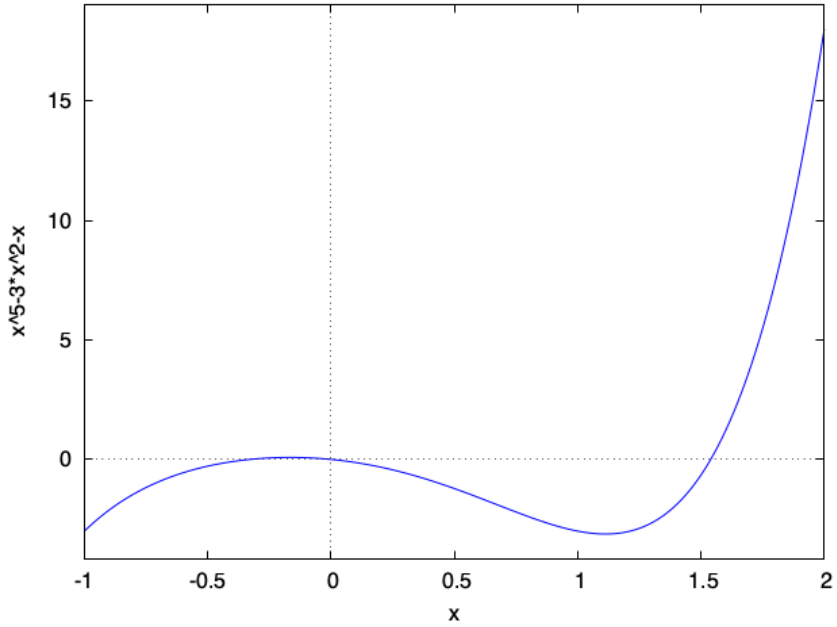
(%i1) `kill(all);`

done (%o1)

(%i2) `f:x^5-3*x^2-x;`

$x^5 - 3\,x^2 - x$ (%o2)

(%i3) `wxplot2d(f,[x,-1,2]);`



(%t3)

(%i4) `max:at(f,x=2);`

18 (%o4)

(%i5) `df:diff(f,x);`

$5\,x^4 - 6\,x - 1$ (%o5)

(%i6) `pc:float(solve(df=0,x));`

$[x = -\left(0.9261\,i\right) - 0.4736, x = 0.9261\,i - 0.4736, x = -0.166, x = 1.113]$ (%o6)

(%i7) `min:at(f,pc[4]);`

-3.121355009085729 (%o7)

(%i8) `izq:at(f,x=-1);`

-3 (%o8)

El mínimo es (1'113,-3'12135) y el máximo (2,18)

2.3. Estudia la función f(x)=(x^2-2x) e^(-x^2/5) en [-2,2].

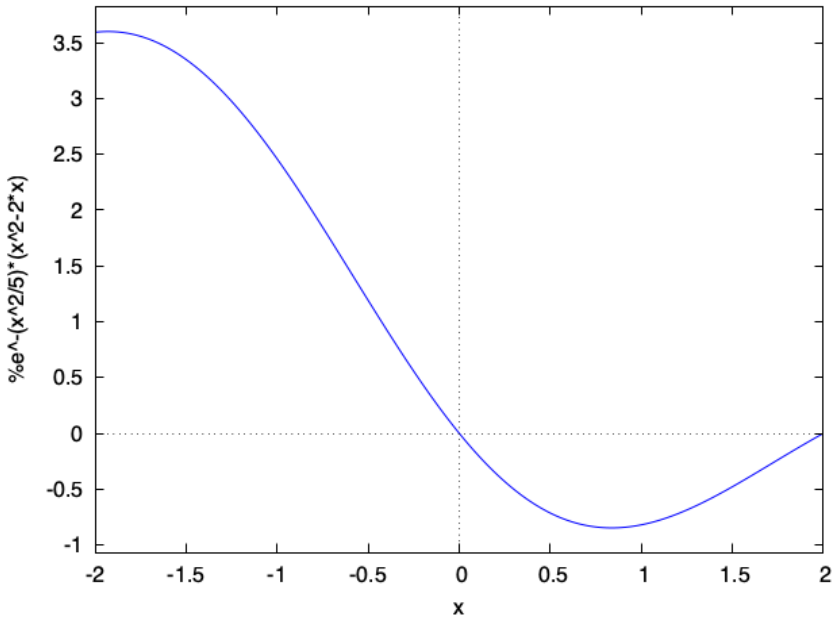
(%i1) `kill(all);`

done (%o1)

(%i2) `f:(x^2-2*x)*%e^(-x^2/5);`

$e^{-\left(\frac{x^2}{5}\right)}\,\left(x^2 - 2\,x\right)$ (%o2)

(%i3) `wxplot2d(f,[x,-2,2]);`



(%t3)

(%i4) `max:float(at(f,x=-2));`

3.5946317129377725 (%o4)

(%i5) `dif:diff(f,x);`

$$e^{-\left(\frac{x^2}{5}\right)}\,(2\,x-2)-\frac{2\,e^{-\left(\frac{x^2}{5}\right)}\,x\,\left(x^2-2\,x\right)}{5} \quad (\%o5)$$

(%i6) `pc:float(solve(dif=0,x));`

$$\left[x=(-(0.866\,i)-0.5)\,(3.02\,i-0.537)^{\frac{1}{3}}+\frac{2.111\,(0.866\,i-0.5)}{(3.02\,i-0.537)^{\frac{1}{3}}}+0.6667,x=(0.866\,i-0.5)\,(3.02\,i-0.537)^{\frac{1}{3}}+\frac{2.111\,(-(0.866\,i)-0.5)}{(3.02\,i-0.537)^{\frac{1}{3}}}+0.6667,x=(3.02\,i-0.537)^{\frac{1}{3}}+\frac{2.111}{(3.02\,i-0.537)^{\frac{1}{3}}}+0.6667\right] \quad (\%o6)$$

(%i7) `pcmejorado:rectform(pc);`

$$\left[x=2.22\times10^{-16}\,i+0.837,x=-\left(1.11\times10^{-16}\,i\right)-1.931,x=3.094\right] \quad (\%o7)$$

Maxima tiene errores de redondeo, si nos fijamos es muy pequeño el valor complejo, por lo que es 0 en realidad, nos las cargamos:

(%i8) `pcsincomplejo:realpart(pcmejorado);`

$[x=0.8370381432224696,x=-1.930801600172758,x=3.093763456950288]$ (%o8)

(%i9) `min:at(f,pcsincomplejo[1]);`

-0.8461639590847368 (%o9)

2.4. Calcula el mínimo absoluto de f(x)= x^2-8sqrt(x)+5. Repite el ejercicio cambiando la raíz cuadrada por la raíz cúbica.

(%i1) `kill(all);`

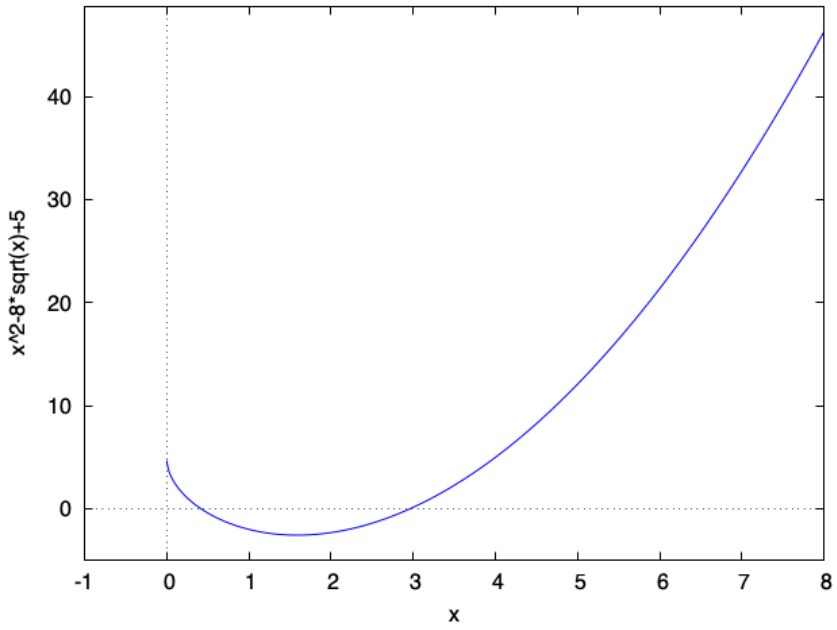
done (%o1)

(%i2) `f:x^2-8*sqrt(x)+5;`

$x^2-8\sqrt{x}+5$ (%o2)

(%i3) `wxplot2d(f,[x,-1,8]);`

```
evaluaciones de expresiones para valor no numérico en algún lugar en el rango
del gráfico.
```



(%t3)

```
(%i4) dif:diff(f,x);
```

$$2\,x-\frac{4}{\sqrt{x}}$$ (%o4)

```
(%i5) pc:float(solve(dif=0,x));
```

$$[x=-\left(0.3969\,\left(3.464\,i+2.0\right)\right),x=0.3969\,\left(3.464\,i-2.0\right),x=1.587]$$ (%o5)

```
(%i6) min:at(f,pc[3]);
```

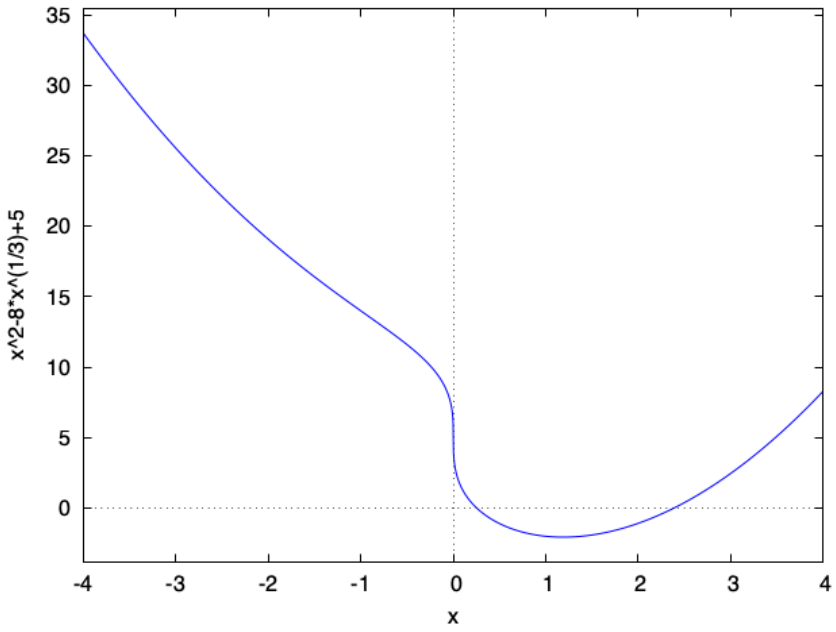
$$-2.5595262993692396$$ (%o6)

Ahora le añadimos la raíz cuadrada:

```
(%i7) g:x^2-8*x^(1/3)+5;
```

$$x^2-8\,x^{\frac{1}{3}}+5$$ (%o7)

```
(%i8) wxplot2d(g,[x,-4,4]);
```



(%t8)

(%i9) `dig:diff(g,x);`

$$2\,x-\frac{8}{3\,x^{\frac{2}{3}}}$$ (%o9)

(%i10) `pcg:float(rectform(solve(dig=0,x)));`

$$[x=-(0.6985\,i)-0.9614,x=1.13\,i+0.3672,x=0.3672-1.13\,i,x=0.6985\,i-0.9614,x=1.188]$$ (%o10)

(%i11) `min:at(g,pcg[5]);`

$$-2.0614922736587493$$ (%o11)

2.5. Un hilo de 10 metros se divide en dos trozos. Con uno de ellos se forma un círculo y con el otro un cuadrado. Halla la forma de hacerlo para que la suma de las áreas de ambas figuras sea mínima.

(%i1) `kill(all);`

done (%o1)

Tenemos los siguientes datos: - x= longitud del trozo para el círculo. - (10-x) = trozo para el cuadrado. - Radio círculo: 2*pi*r = x –> r = x / (2*pi) - Lado cuadrado : 4*l = 10-x –> l = (10-x)/4

(%i2) `area_circulo: %pi * (x / (2*%pi))^2;`

$$\frac{x^2}{4\,\pi}$$ (%o2)

(%i3) `area_cuadrado: ((10-x)/4)^2;`

$$\frac{(10-x)^2}{16}$$ (%o3)

(%i4) `f: area_circulo + area_cuadrado;`

$$\frac{x^2}{4\,\pi}+\frac{(10-x)^2}{16}$$ (%o4)

Buscamos ahora el mínimo en [0,10]

(%i5) `df: diff(f, x);`

$$\frac{x}{2\,\pi}-\frac{10-x}{8} \qquad (\%o_5)$$

(%i6) `pc: float(solve(df=0, x));`

$$[x = 4.399008464884426] \qquad (\%o_6)$$

Ahora para asegurarnos que es un mínimo comparamos los extremos:

(%i7) `float([at(f, x=0), at(f, pc[1]), at(f, x=10)]);`

$$[6.25, 3.5006197094472338, 7.957747154594768] \qquad (\%o_7)$$

El mínimo se alcanza en el punto crítico (aprox x=4.4m para el círculo)

2.6. Define la función a trozos f(x)=x^2 si x<0, x^3 si x>0 y dibuja su gráfica.

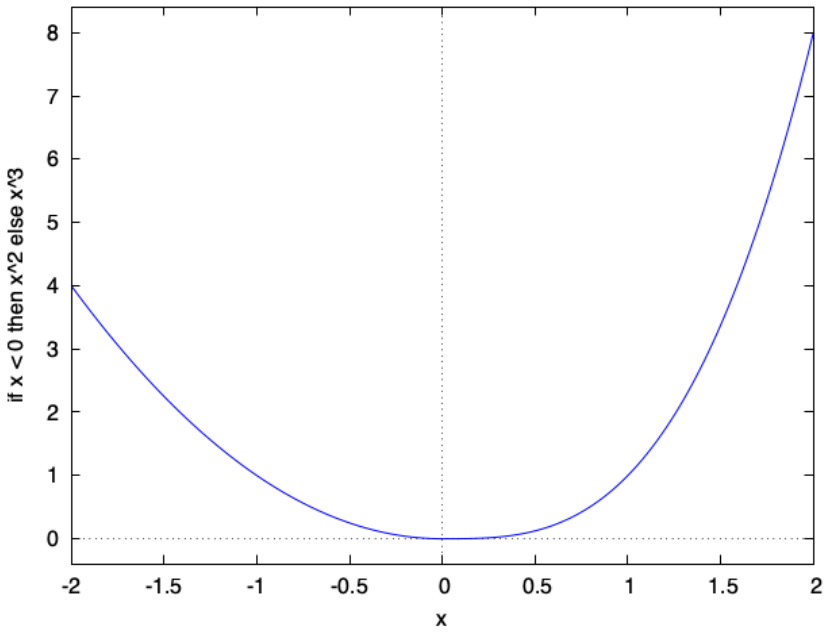
(%i1) `kill(all);`

done (%o1)

(%i2) `f: if x<0 then x^2 else x^3;`

if $x < 0$ **then** x^2 **else** x^3 (%o2)

(%i3) `wxplot2d(f, [x,-2,2]);`



(%t3)

2.7. Calcula los extremos absolutos de f(x)= 2x^3-3x-1 si x<0 (x^2 -1)e^(-x) x>0 en el intervalo [-2,2]

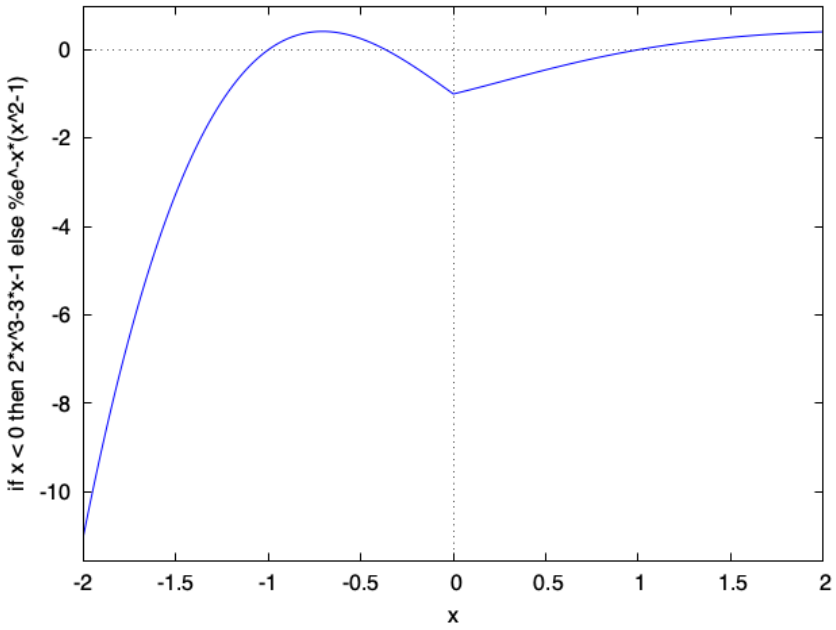
(%i1) `kill(all);`

done (%o1)

(%i2) `f: if x<0 then 2*x^3-3*x-1 else (x^2-1)*%e^(-x);`

if $x < 0$ **then** $2\,x^3 - 3\,x - 1$ **else** $e^{-x}\left(x^2-1\right)$ (%o2)

(%i3) `wxplot2d(f, [x,-2,2]);`



(%t3)

Para hallar los extremos estudiamos cada trozo por separado:

(%i4) `f_izq: 2*x^3-3*x-1;`

$$2\,x^3 - 3\,x - 1$$ (%o4)

(%i5) `f_der: (x^2-1)*%e^(-x);`

$$e^{-x}\left(x^2-1\right)$$ (%o5)

(%i6) `df_izq: diff(f_izq, x);`

$$6\,x^2 - 3$$ (%o6)

(%i7) `df_der: diff(f_der, x);`

$$2\,e^{-x}\,x - e^{-x}\left(x^2-1\right)$$ (%o7)

Ahora calculamos los puntos donde f ' (x) en cada intervalo

(%i8) `pc_izq: float(solve(df_izq=0, x));`

$$[x = -0.7071067811865475, x = 0.7071067811865475]$$ (%o8)

(%i9) `pc_der: float(solve(df_der=0, x));`

$$[x = -0.41421356237309515, x = 2.414213562373095]$$ (%o9)

Por tanto de la parte izquierda nos quedamos el punto x=-0.7071067811865475 y de la parte derecha no valen, por tanto:

(%i10) `maximo:float(at(f_izq,pc_izq[1]));`

0.41421356237309503 (%o10)

2.8. Calcula el polinomio de Taylor de f(x)=cos(x) de grado 8 centrado en x=0. Dibuja conjuntamente la gráfica de f y del polinomio de Taylor.

(%i1) `kill(all);`

done (%o1)

(%i2)
```
f: cos(x);
```
$\cos x$ (%o2)

(%i3)
```
T: taylor(f, x, 0, 8);
```
$$1 - \frac{x^2}{2} + \frac{x^4}{24} - \frac{x^6}{720} + \frac{x^8}{40320} + \cdots$$ (%o3)

(%i4)
```
wxplot2d([f, T], [x, -4, 4], [style, [lines, 2], [lines, 2]]);
```
$$wxplot2d\left(\left[\cos x, 1 - \frac{x^2}{2} + \frac{x^4}{24} - \frac{x^6}{720} + \frac{x^8}{40320} + \cdots\right], [x, -4, 4], [style, [lines, 2], [lines, 2]]\right)$$ (%o4)

CAPÍTULO 3

Práctica 3: Programación, iteración y método de Newton

Resumen de la práctica

Tercera práctica centrada en la **programación elemental** con Maxima (bucles `for` y condicionales `if`), el estudio de **ecuaciones en diferencias** y la implementación numérica del **método de Newton** para aproximar ceros de funciones.

3.1. Programación básica

Las dos estructuras elementales de programación que vamos a considerar son los bucles y los condicionales. Veamos cómo se implementan en MAXIMA.

El comando que vamos a usar para definir bucles es el comando for en combinación con otros comandos subsidiarios. Analicemos el siguiente micro-programa.

```
(%i1) for i:1 thru 26 step 4 do print(i^2);
```

```
     1   25   81   169   289   441   625
```

done (%o1)

En la anterior entrada

for establece la estructura del bucle i:1 establece el contador i y le da el valor inicial 1 thru 26 dice que el contador va a tomar valores solo hasta 26 step 4 dice que el valor del contador se incrementará 4 unidades cada vez que se ejecute. do print(i^2) establece la acción a realizar en cada paso del bucle; en este caso se muestra el cuadrado del contador.

Nota

Cuando el paso (step) sea 1 no es necesario escribirlo.

Ejercicio

Con un bucle resuelva la ecuación en diferencias x_{n+1}=x_n+0.01(20-x_n^2) hasta la iterada número 40, empezando en x_0=0

```
(%i2) g:x+0.01*(20-x^2);
```

$$0.01\left(20-x^2\right)+x \qquad (\%o_2)$$

```
(%i3) a[0]:0;
```

$$0 \qquad (\%o_3)$$

```
(%i4) for i: 1 thru 40 do (    a[i]:at(g,x=a[i-1]),display(a[i]) );
```

```
     a[1] = 0.2   a[2] = 0.3996   a[3] = 0.5980031984   a[4] = 0.7944271201470338   a[5]
     = 0.9881159756547827   a[6] = 1.1783522438413407   a[7] = 1.3644671037356815
     a[8] = 1.545849398963913   a[9] = 1.721952895321142   a[10] = 1.8923016775840935
```

```
a[11] = 2.056493621194218  a[12] = 2.214201961054093  a[13] = 2.365175057810735
a[14] = 2.509234527269835  a[15] = 2.6462719481414045  a[16] =
2.7762443959062035  a[17] = 2.8991690664481977  a[18] = 3.0151172536896964
a[19] = 3.1242079331547234  a[20] = 3.226601181058854  a[21] = 3.32249162924275
a[22] = 3.412102122978869  a[23] = 3.4956777140025  a[24] = 3.5734800872007626
a[25] = 3.645782487864559  a[26] = 3.71286518837636  a[27] = 3.77501150930579
a[28] = 3.8325043903518785  a[29] = 3.885623491331214  a[30] =
3.9346427921673643  a[31] = 3.9798286531478184  a[32] = 4.021438292063655
a[33] = 4.059718632694897  a[34] = 4.094905478928395  a[35] =
4.1272229701148175  a[36] = 4.156883275664383  a[37] = 4.184086489989401  a[38]
= 4.209020692432283  a[39] = 4.231862140539051  a[40] = 4.252775568773774
```

done (%o4)

(%i5) `g:cos(x);`

$\cos x$ (%o5)

(%i6) `for i:1 thru 60 do (  a[i]:float(at(g,x=a[i-1])),display(a[i]) );`

```
a[1] = 1.0  a[2] = 0.5403023058681398  a[3] = 0.8575532158463933  a[4] =
0.6542897904977792  a[5] = 0.7934803587425655  a[6] = 0.7013687736227566  a[7]
= 0.7639596829006542  a[8] = 0.7221024250267077  a[9] = 0.7504177617637605
a[10] = 0.7314040424225098  a[11] = 0.7442373549005569  a[12] =
0.7356047404363473  a[13] = 0.7414250866101093  a[14] = 0.7375068905132428
a[15] = 0.7401473355678757  a[16] = 0.7383692041223232  a[17] =
0.739567202212256  a[18] = 0.7387603198742114  a[19] = 0.7393038923969057
a[20] = 0.7389377567153446  a[21] = 0.7391843997714936  a[22] =
0.7390182624274122  a[23] = 0.7391301765296711  a[24] = 0.7390547907469175
a[25] = 0.7391055719265361  a[26] = 0.739071365298945  a[27] =
0.7390944073790912  a[28] = 0.7390788859949922  a[29] = 0.7390893414033927
a[30] = 0.7390822985224023  a[31] = 0.7390870426953322  a[32] =
0.7390838469650002  a[33] = 0.7390859996481299  a[34] = 0.7390845495752126
a[35] = 0.7390855263619245  a[36] = 0.7390848683867142  a[37] =
0.7390853116067619  a[38] = 0.7390850130484203  a[39] = 0.7390852141609171
a[40] = 0.7390850786891229  a[41] = 0.7390851699445545  a[42] =
0.7390851084737986  a[43] = 0.7390851498812395  a[44] = 0.7390851219886894
a[45] = 0.7390851407774467  a[46] = 0.7390851281211138  a[47] =
0.7390851366465718  a[48] = 0.7390851309037207  a[49] = 0.7390851347721744
a[50] = 0.7390851321663374  a[51] = 0.7390851339216605  a[52] =
0.7390851327392538  a[53] = 0.7390851335357372  a[54] = 0.7390851329992164
a[55] = 0.7390851333606232  a[56] = 0.7390851331171753  a[57] = 0.7390
...(salida truncada)
```

done (%o6)

En cuanto a los condicionales, el comando que utilizaremos será if. Su presentación más básica es de la forma if condicion1 then accion1 else accion2

La salida de este comando será acción 1 si se cumple condicion1 y accion 2 si no se cumple condicion1. Veamos algunos ejemplos.

(%i7) `f(x):=x^3-8;`

$f(x) := x^3 - 8$ (%o7)

(%i8) `valor:-2;`

-2 (%o8)

```
(%i9) if f(valor)<0 then  print("El valor de f en ", valor ," es negativo")  else print(
"El valor de f en ", valor," no es negativo");

     El valor de f en -2 es negativo
```

es negativo (%o9)

Con el comando if también se pueden considerar múltiples condiciones. Observa el siguiente ejemplo.

```
(%i10) valor:3;
```

3 (%o10)

```
(%i11) if f(valor)<0 then print("Da negativo") elseif f(valor)=0 then print("Da cero")
 elseif f(valor)>0 then print("Da positivo");

     Da positivo
```

Da positivo (%o11)

Con el comando if se puede establecer una condición para salir de un bucle. El comando de salida que utilizaremos es return(arg). El argumento de return es un valor que se muestra al salir del bucle. Si no queremos mostrar nada usaremos convenientemente el símbolo $.

Ejercicio

Escribe un programa que evalúe la función f(x)=sen(x/24) en los 100 primeros números naturales, pero que tan pronto uno de esos valores sea negativo se muestre en pantalla y se termine la evaluación.

```
(%i12) f(x):=sin(x/24);
```

$$f\left(x\right):=\sin\left(\frac{x}{24}\right)$$ (%o12)

```
(%i13) for i:1 thru 100 do (    val: f(i),    if val < 0 then (        print("Valor negativo
encontrado en i=", i, ": ", val),        return(done)    ) );

     Valor negativo encontrado en i= 76 : sin(19/6)
```

done (%o13)

3.2. Iteración de funciones

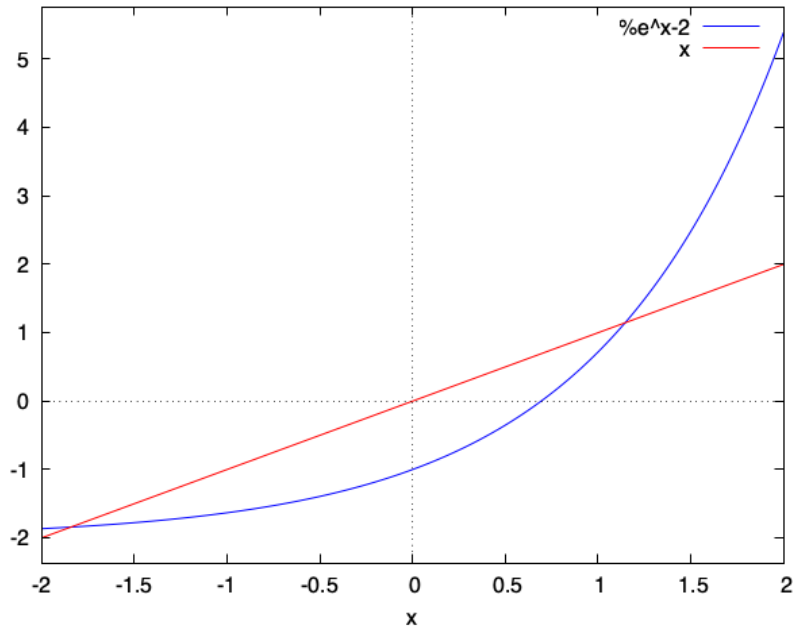
Vamos a aplicar los bucles (y los condicionales) al cálculo de iteradas de funciones.

Consideremos el problema x_(n+1)= e^(x_n)-2, es decir la ecuación en diferencias asociada a g(x)=e^x-1, en el intervalo [-2,2].

```
(%i1)  g:%e^x-2;
```

$e^{x}-2$ (%o1)

```
(%i2)  wxplot2d([g,x],[x,-2,2]);
```



(%t2)

Vemos gráficamente que tiene dos puntos fijos. Comprobémoslo analíticamente.

(%i3)
```
f:x-g;
```

$x - e^x + 2$ (%o3)

(%i4)
```
df:diff(f,x);
```

$1 - e^x$ (%o4)

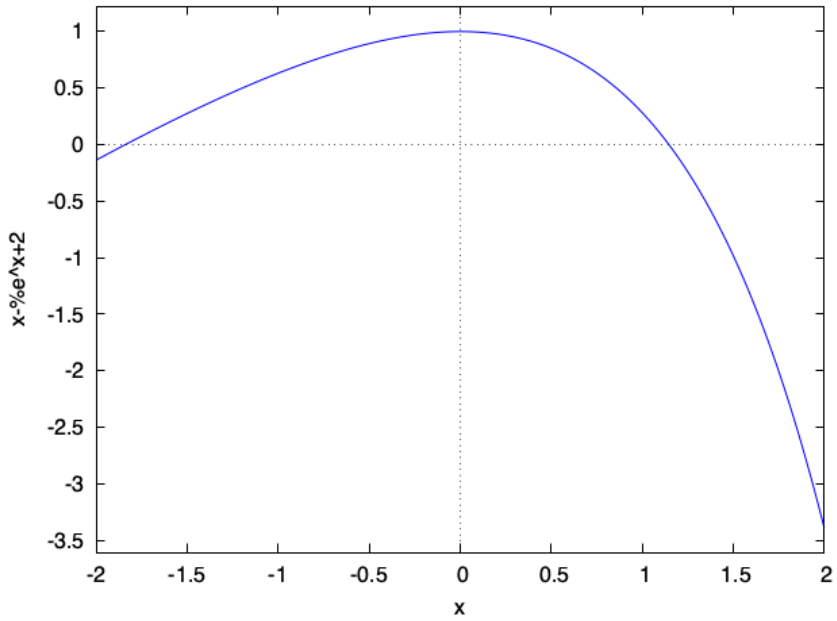
(%i5)
```
s:solve(df=0,x);
```

$[x = 0]$ (%o5)

(%i6)
```
valores:float([at(f,x=-2),at(f,s[1]),at(f,x=2)]);
```

$[-0.1353352832366127, 1.0, -3.3890560989306504]$ (%o6)

(%i7)
```
wxplot2d(f,[x,-2,2]);
```



(%t7)

Veamos cuál de ellos es atractor.

El de la izquierda está situado en [-2,-1.5]. Estudiemos la función g ahí.

```
(%i8) dg:diff(g,x);
```

$$e^x \qquad (\%o_8)$$

```
(%i9) ddg:diff(g,x,2);
```

$$e^x \qquad (\%o_9)$$

La segunda derivada es positiva, por tanto g es convexa. En particular, los valores extremos de su derivada se alcanzan en los extremos del intervalo.

```
(%i10) derivadas:float([at(dg,x=-2),at(dg,x=-1.5)]);
```

$$[0.1353352832366127, 0.22313016014842982] \qquad (\%o_{10})$$

Como la derivada está por tanto estrictamente entre -1 y 1, ese punto es atractor.

Comprobemos que el intervalo [-2,-1.5] se transforma por g en sí mismo.

```
(%i11) valoresg:float([at(g,x=-2),at(g,x=-1.5)]);
```

$$[-1.8646647167633872, -1.77686983985157] \qquad (\%o_{11})$$

Evidentemente g([-2,-1.5]) está contenido en [-1.86,-1.77], que está contenido en [-2,-1.5]

Definamos las iteradas y veamos que convergen.

```
(%i12) a[0]:-1.8;
```

$$-1.8 \qquad (\%o_{12})$$

```
(%i13) for i:1 thru 20 do (a[i]:float(at(g,x=a[i-1])),display(a[i]));
```

```
                a[1] = -1.8347011117784136   a[2] = -1.840338784506017   a[3] =
-1.8412363696632286   a[4] = -1.8413788096057715   a[5] = -1.8414014019899503
 a[6] = -1.8414049850699314   a[7] = -1.8414055533275364   a[8] =
-1.8414056434500115   a[9] = -1.8414056577429267   a[10] = -1.8414056600097022
 a[11] = -1.8414056603692   a[12] = -1.8414056604262141   a[13] =
-1.8414056604352562   a[14] = -1.8414056604366904   a[15] = -1.8414056604369178
 a[16] = -1.8414056604369538   a[17] = -1.8414056604369595   a[18] =
-1.8414056604369604   a[19] = -1.8414056604369606   a[20] = -1.8414056604369606
```

done $(\%o_{13})$

Estudiemos el punto fijo de la derecha.

De nuevo, como g es convexa, estudiamos la derivada en los extremos de [1,1.5], donde está situada el punto fijo.

```
(%i14) vg:float([at(dg,x=1),at(dg,x=1.5)]);
```

$$[2.718281828459045, 4.4816890703380645] \qquad (\%o_{14})$$

Evidentemente, la derivada en este caso no está entre -1 y 1. No es atractor. Si se quiere calcular este punto fijo, ha de utilizarse otro método, como el método de Newton.

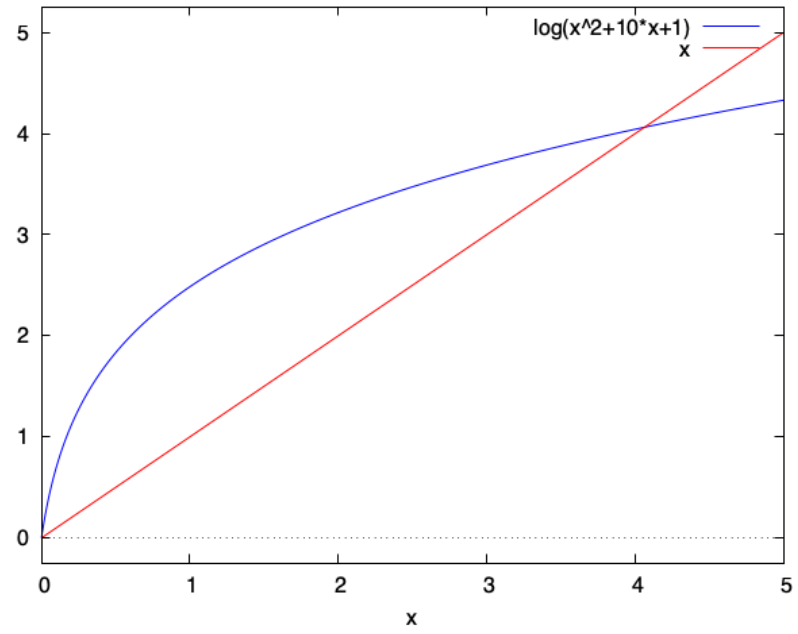
Ejercicio

Estudia las iteraciones x_(n+1)=g(x_n) donde g(x)=log(x^2+10x+1) en x>=0.

```
(%i15) g: log(x^2+10*x+1);
```

$$\log\left(x^2 + 10\,x + 1\right) \qquad (\%o_{15})$$

```
(%i16) wxplot2d([g, x], [x, 0, 5]);
```



(%t16)

(%i17) a[0]: 1.0;

1.0 (%o17)

(%i18) for i:1 thru 20 do (a[i]: float(at(g, x=a[i-1])), display(a[i]));

```
a[1] = 2.4849066497880004  a[2] = 3.4664802368406704  a[3] = 3.8645390274430866
a[4] = 3.999668471107175  a[5] = 4.042946570947716  a[6] = 4.056554125609098
a[7] = 4.060807989279908  a[8] = 4.062135395217307  a[9] = 4.062549375450611
a[10] = 4.062678461485154  a[11] = 4.062718710490893  a[12] = 4.062731259910645
a[13] = 4.062735172730247  a[14] = 4.062736392717477  a[15] = 4.06273677309997
a[16] = 4.062736891700242  a[17] = 4.062736928678873  a[18] = 4.062736940208518
a[19] = 4.062736943803371  a[20] = 4.062736944924217
```

done (%o18)

3.3. El método de Newton para aproximar ceros de funciones

Pasemos al método de Newton.

Resolvamos la ecuación x^2=sen(x)+1 en [0, pi]

(%i1) kill(all);

done (%o1)

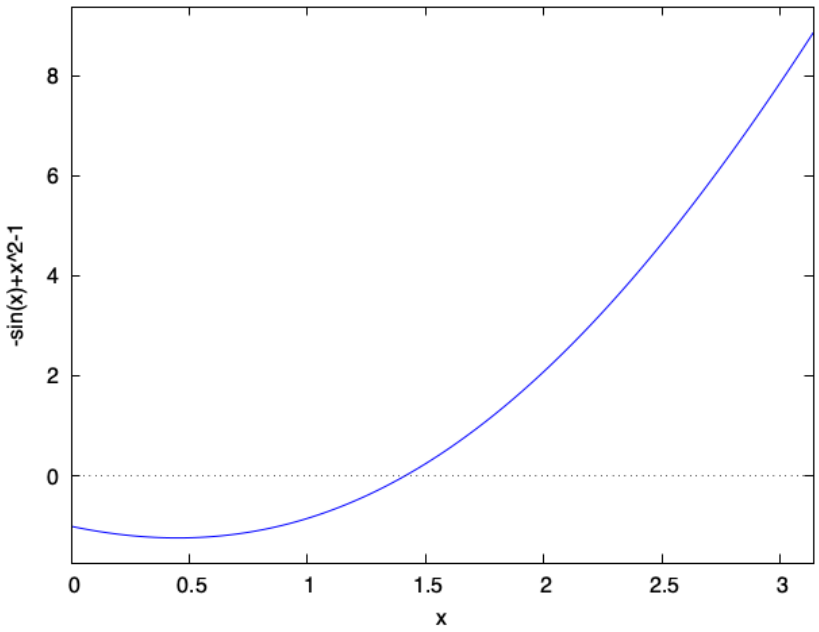
(%i2) f:x^2-sin(x)-1;

$-\sin x + x^2 - 1$ (%o2)

(%i3) df:diff(f,x);

$2\,x - \cos x$ (%o3)

(%i4) wxplot2d(f,[x,0,%pi]);



(%t4)

Una programación del método de Newton puede ser:

```
(%i5) a[0]:1.5;
```

1.5 (%o5)

```
(%i6) for i:1 thru 10 do ( a[i]: float(a[i-1]-at(f,x=a[i-1])/at(df,x=a[i-1])), display(
a[i]) );
```

```
a[1] = 1.4137991259986291   a[2] = 1.4096337516523312   a[3] = 1.4096240040559695
a[4] = 1.4096240040025962   a[5] = 1.4096240040025962   a[6] = 1.4096240040025962
a[7] = 1.4096240040025962   a[8] = 1.4096240040025962   a[9] = 1.4096240040025962
a[10] = 1.4096240040025962
```

done (%o6)

Cambiemos el programa para que nos muestre sólo un resultado final una vez impuesta una condición de parada. Por ejemplo, que la diferencia entre dos iteradas consecutivas sea menor que un cierto numero llamado tol.

```
(%i7) a[0]:1.5;
```

1.5 (%o7)

```
(%i8) tol:10^(-10);
```

$$\frac{1}{10000000000}$$ (%o8)

```
(%i9) for i:1 thru 20 do ( a[i]: float(a[i-1]-at(f,x=a[i-1])/at(df,x=a[i-1])), if abs(
a[i]-a[i-1])<tol then (print("Una aproximación buena de la raíz es ",a[i], "alcanzada
en la iterada número ",i),return(done)) );
```

```
Una aproximación buena de la raíz es 1.4096240040025962 alcanzada en la iterada
número 4
```

done (%o9)

Finalmente, reescribamos este programa como una función cuyos argumentos sean la función a la cual le queremos aproximar la raíz y la condición inicial. La tolerancia la fijamos en 10^(-10) .

```
(%i10) newton(funcion,condini):=block([valorfun,derfuncion,valorder,a], a[0]:condini, for
i:1 thru 20 do (valorfun:at(funcion,x=a[i-1]), derfuncion: diff(funcion,x), valorder:at(
```

```
derfuncion,x=a[i-1]), a[i]:float( a[i-1]-valorfun/valorder),  if abs(a[i]-a[i-1])<10^(
-10) then (print("Una aproximación buena de la raíz es ",a[i],  "alcanzada en la iterada
número ",i),return(done)) ) );
```

(%o10)

(%i11) `newton(x^2-2,3);`

```
Una aproximación buena de la raíz es 1.414213562373095 alcanzada en la iterada
número 6
```

done (%o11)

(%i12) `newton(sin(x),3.0);`

```
Una aproximación buena de la raíz es 3.141592653589793 alcanzada en la iterada
número 4
```

done (%o12)

(%i13) `newton(cos(x)-x,0.0);`

```
Una aproximación buena de la raíz es 0.7390851332151607 alcanzada en la iterada
número 6
```

done (%o13)

(%i14) `newton(exp(x)+x,0.0);`

```
Una aproximación buena de la raíz es -0.567143290409784 alcanzada en la iterada
número 5
```

done (%o14)

Ejercicio

Calcula de manera aproximada los puntos donde se anula f(x)=cos(3x)-5sen(x)+x en [0, pi]. Calcula también su mínimo.

(%i15) `f:cos(3*x)-5*sin(x)+x;`

$$\cos(3\,x) - 5\,\sin x + x$$ (%o15)

(%i16) `a[0]:0.2;`

0.2 (%o16)

(%i17) `for i:1 thru 10 do (a[i]: float(a[i-1]-at(f,x=a[i-1])/at(df,x=a[i-1])), display(
a[i]));`

```
a[1] = 0.25514705062550125  a[2] = -0.36959520704499316  a[3] =
0.7564346904447391  a[4] = 4.980444371358241  a[5] = 4.043810627196813  a[6] =
3.024662442471924  a[7] = 2.8113515108103915  a[8] = 2.7136283611774634  a[9] =
2.657522457419028  a[10] = 2.623283537437053
```

done (%o17)

(%i18) `a[0]:2.5;`

2.5 (%o18)

(%i19) `for i:1 thru 10 do (a[i]: float(a[i-1]-at(f,x=a[i-1])/at(df,x=a[i-1])), display(
a[i]));`

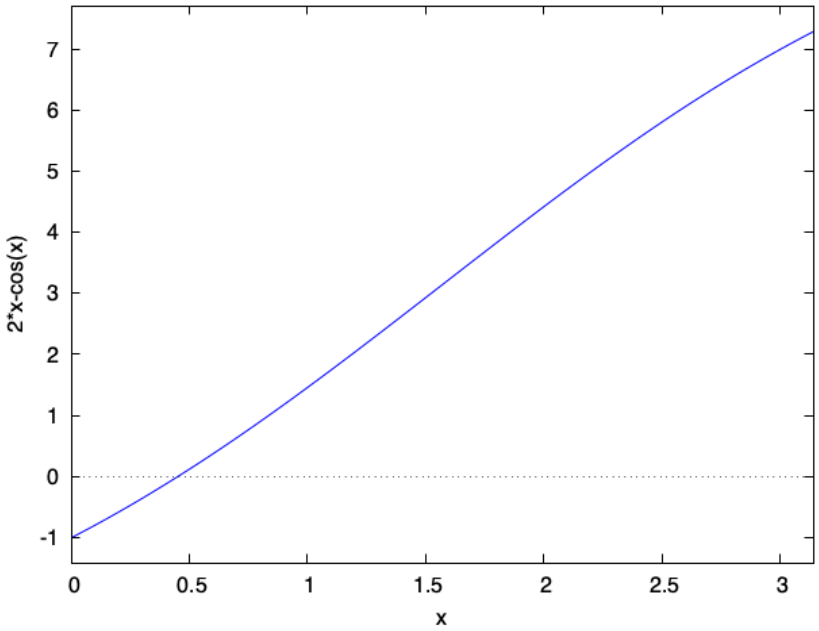
```
a[1] = 2.5251201163670927  a[2] = 2.5405735983088986  a[3] = 2.5501681234620213
a[4] = 2.55615019792147  a[5] = 2.5598877066972854  a[6] = 2.56222538702425
a[7] = 2.563688400870152  a[8] = 2.564604328359309  a[9] = 2.56517786600784
a[10] = 2.5655370491198366
```

done (%o19)

(%i20) `dff:diff(df,x);`

$\sin x + 2$ (%o20)

(%i21) `wxplot2d(df,[x,0,%pi]);`



(%t21)

(%i22) `a[0]:1;`

1 (%o22)

(%i23) `for i:1 thru 10 do ( a[i]: float(a[i-1]-at(df,x=a[i-1])/at(dff,x=a[i-1])), display(a[i]) );`

```
a[1] = 0.4862880170389824  a[2] = 0.45041860473199363  a[3] =
0.45018362150211116  a[4] = 0.4501836112948736  a[5] = 0.45018361129487355
a[6] = 0.45018361129487355  a[7] = 0.45018361129487355  a[8] =
0.45018361129487355  a[9] = 0.45018361129487355  a[10] = 0.45018361129487355
```

done (%o23)

Dado la función y=cosx, calcula el rectángulo con mayor y menor diagonal (con los lados paralelos a los ejes)

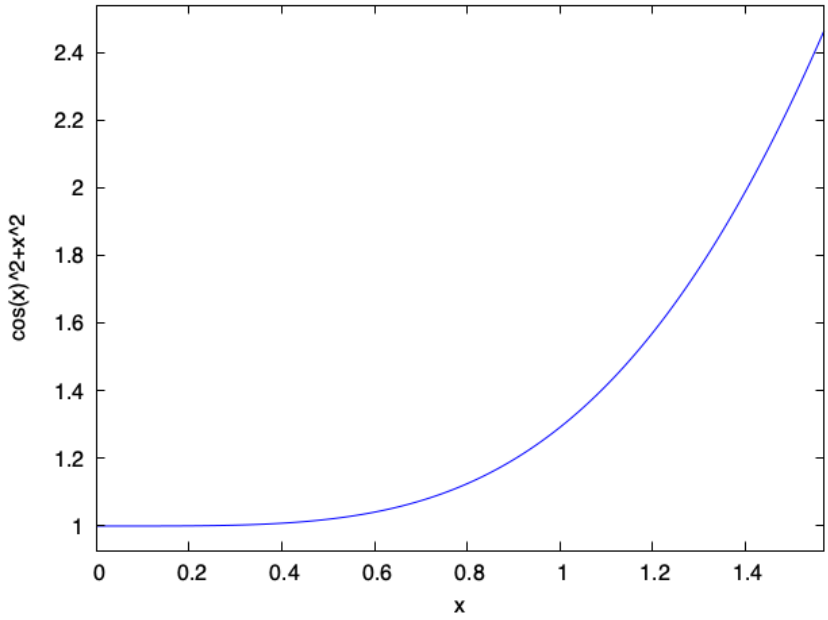
(%i24) `h:x^2+(cos(x))^2;`

$\cos^2 x + x^2$ (%o24)

(%i25) `dh:diff(h,x);`

$2\,x - 2\,\cos x\,\sin x$ (%o25)

(%i26) `wxplot2d(h,[x,0,%pi/2]);`



(%t26)

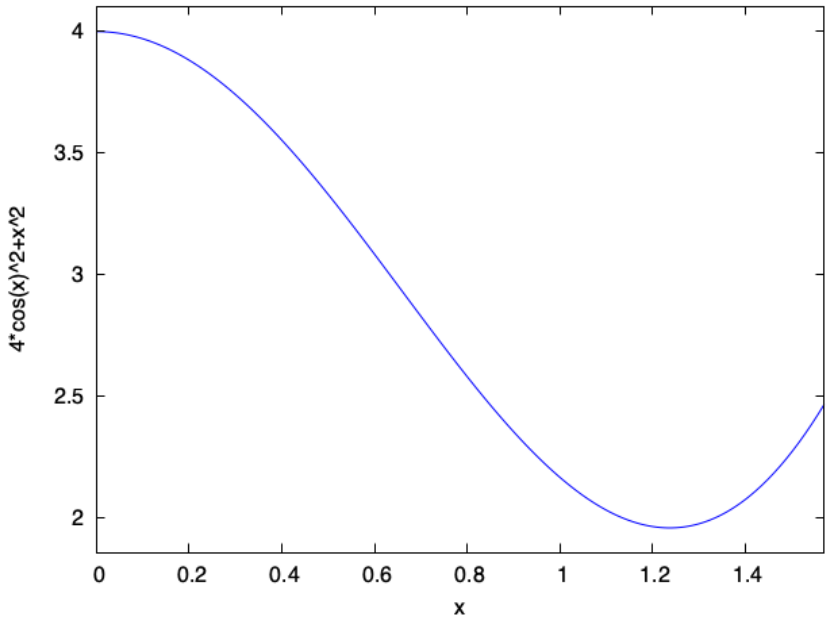
No tiene, son los extremos

Dado la función y=2cosx, calcula el rectángulo con mayor y menor diagonal (con los lados paralelos a los ejes)

(%i27) `f:x^2+4*(cos(x))^2;`

$4\cos^2 x + x^2$ (%o27)

(%i28) `wxplot2d(f,[x,0,%pi/2]);`



(%t28)

(%i29) `ddf:diff(df,x);`

$\sin x + 2$ (%o29)

(%i30) `a[0]:1.2;`

1.2 (%o30)

(%i31) `for i:1 thru 10  do ( a[i]: float(a[i-1]-at(df,x=a[i-1])/at(ddf,x=a[i-1])), display(a[i]) );`

```
a[1] = 0.5050426048972178   a[2] = 0.450719020462328   a[3] = 0.45018366427074064
a[4] = 0.4501836112948741   a[5] = 0.45018361129487355   a[6] =
0.45018361129487355   a[7] = 0.45018361129487355   a[8] = 0.45018361129487355
a[9] = 0.45018361129487355   a[10] = 0.45018361129487355
```

done (%o31)

(%i32) `minimo:at(f,x=a[10]);`

3.4453098259343946 (%o32)

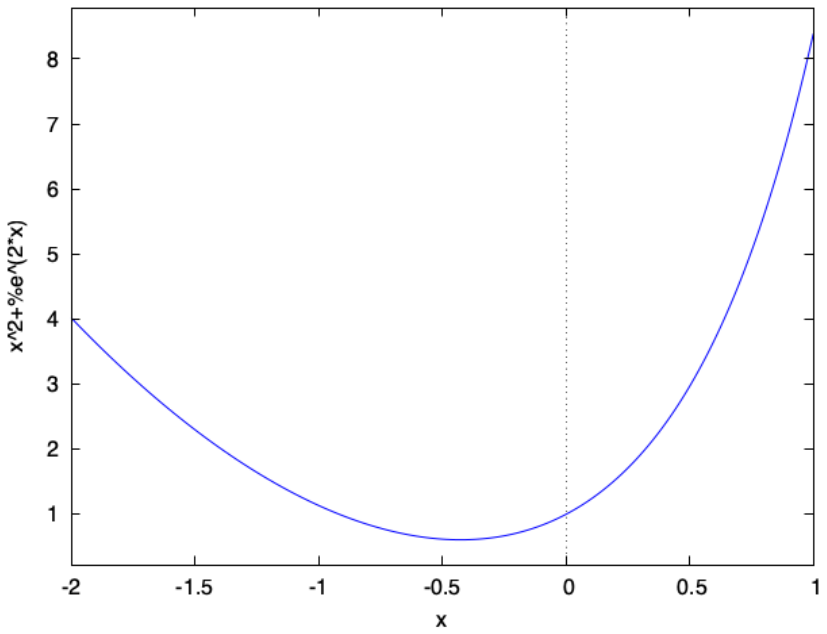
Ejercicio

Distancia mínima a una función exponencial y=e^x al (0,0)

(%i33) `f:x^2+(%e)^(2*x);`

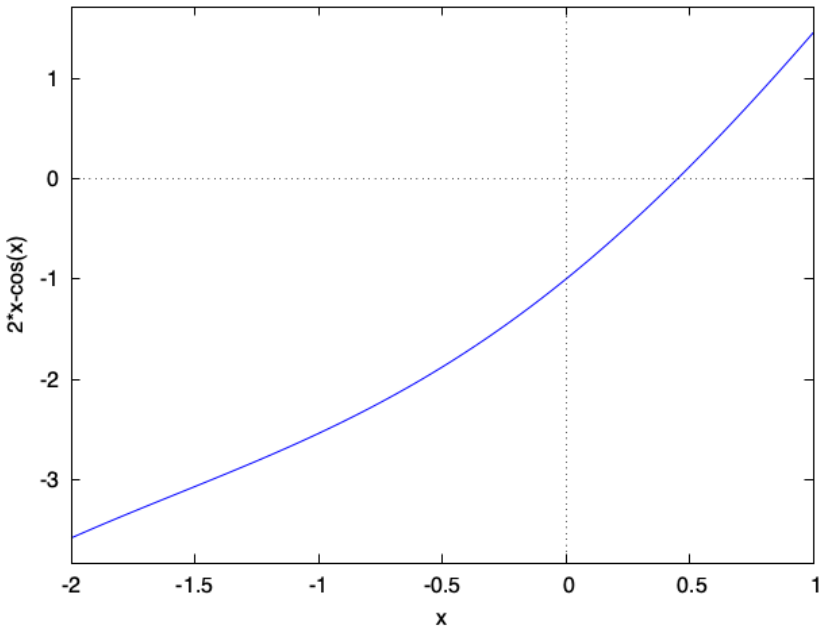
$$x^2 + e^{2\,x}$$ (%o33)

(%i34) `wxplot2d(f,[x,-2,1]);`



(%t34)

(%i35) `wxplot2d(df,[x,-2,1]);`



(%t35)

(%i36) a[0]:-0.5;

-0.5 (%o36)

(%i37) for i:1 thru 10 do (a[i]: float(a[i-1]-at(df,x=a[i-1])/at(ddf,x=a[i-1])), display(a[i]));

```
a[1] = 0.7347850168198033  a[2] = 0.4623211176508535  a[3] =
0.45021061788317746  a[4] = 0.45018361142970736  a[5] = 0.45018361129487355
a[6] = 0.45018361129487355  a[7] = 0.45018361129487355  a[8] =
0.45018361129487355  a[9] = 0.45018361129487355  a[10] = 0.45018361129487355
```

done (%o37)

(%i38) distancia_minima: sqrt(at(f, x=a[10]));

1.6319227257200946 (%o38)

CAPÍTULO 4

Práctica 4: Matrices, vectores y subespacios

Resumen de la práctica

Cuarta práctica dedicada al álgebra lineal computacional: definición y manipulación de matrices, productos, determinantes, rangos, inversas, generación de matrices a partir de fórmulas, y aplicación a la resolución de problemas geométricos en $\mathbb{R}^3$ y $\mathbb{R}^4$.

4.1. Definición de matrices y operaciones básicas.

Empecemos tratando cómo se introducen los vectores y las matrices.

En general, el comando que declara una matriz es el comando matrix. Las filas de la matriz se introducen como listas. Veamos algunos ejemplos

```
(%i1) m:matrix([1,3],[2,-2]);
```

$$\begin{pmatrix} 1 & 3 \\ 2 & -2 \end{pmatrix} \qquad (\%o_1)$$

Extraigamos algunos coeficientes de la matriz m.

```
(%i2) m[1,2];
```

$$3 \qquad (\%o_2)$$

```
(%i3) m[2,2];
```

$$-2 \qquad (\%o_3)$$

Para extraer filas completas tecleamos

```
(%i4) m[1];
```

$$[1,3] \qquad (\%o_4)$$

```
(%i5) m[2];
```

$$[2,-2] \qquad (\%o_5)$$

Definamos otra matriz de orden 2 y sumémoslas.

```
(%i6) n:matrix([2,-3],[2,-1]);
```

$$\begin{pmatrix} 2 & -3 \\ 2 & -1 \end{pmatrix} \qquad (\%o_6)$$

```
(%i7) m+n;
```

$$\begin{pmatrix} 3 & 0 \\ 4 & -3 \end{pmatrix} \qquad (\%o_7)$$

Para multiplicar las matrices hay que utilizar el punto .

(%i8) `n.m;`

$$\begin{pmatrix} -4 & 12 \\ 0 & 8 \end{pmatrix}$$ (%o8)

Ejercicio

Define las matrices m=((1,2,0),(-1,2,3),(1,1,1)) y n=((1,2),(3,3),(4,-2)), y halla su producto.

(%i9) `m:matrix([1,2,0],[-1,2,3],[1,1,1]);`

$$\begin{pmatrix} 1 & 2 & 0 \\ -1 & 2 & 3 \\ 1 & 1 & 1 \end{pmatrix}$$ (%o9)

(%i10) `n:matrix([1,2],[3,3],[4,-2]);`

$$\begin{pmatrix} 1 & 2 \\ 3 & 3 \\ 4 & -2 \end{pmatrix}$$ (%o10)

(%i11) `m.n;`

$$\begin{pmatrix} 7 & 8 \\ 17 & -2 \\ 8 & 3 \end{pmatrix}$$ (%o11)

Pero al revés no se puede

Vamos a definir un vector u y a multiplicarlo por la matriz n.

El vector lo vamos a definir como una lista, no utilizamos el comando matrix para el.

(%i12) `u:[1,3];`

$[1, 3]$ (%o12)

(%i13) `n.u;`

$$\begin{pmatrix} 7 \\ 12 \\ -2 \end{pmatrix}$$ (%o13)

Observa que, aunque escrito como un vector fila, el producto de m por u se hace escribiéndolo como un vector columna.

El punto también sirve para definir el producto escalar de vectores.

(%i14) `v:[2,3];`

$[2, 3]$ (%o14)

(%i15) `u.v;`

11 (%o15)

(%i16) `v.u;`

11 (%o16)

Ejercicio

Define la matriz A=((1,2,3),(3,2,1),(1,-1,1)) y el vector u=(2,-1,1). Llama v al producto Au y calcula el producto escalar de u y v.

(%i17) `A:matrix([1,2,3],[3,2,1],[1,-1,1]);`

$$\begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \\ 1 & -1 & 1 \end{pmatrix} \qquad (\%o_{17})$$

```
(%i18) u:[2,-1,1];
```

$$[2,-1,1] \qquad (\%o_{18})$$

```
(%i19) v:A.u;
```

$$\begin{pmatrix} 3 \\ 5 \\ 4 \end{pmatrix} \qquad (\%o_{19})$$

El determinante y el rango de una matriz tienen sus propios comandos.

```
(%i20) m:matrix([1,1,1],[2,1,1],[0,1,-2]);
```

$$\begin{pmatrix} 1 & 1 & 1 \\ 2 & 1 & 1 \\ 0 & 1 & -2 \end{pmatrix} \qquad (\%o_{20})$$

```
(%i21) determinant(m);
```

$$3 \qquad (\%o_{21})$$

```
(%i22) rank(m);
```

$$3 \qquad (\%o_{22})$$

```
(%i23) rank(matrix([1,2,2],[2,1,1],[-1,1,1]));
```

$$2 \qquad (\%o_{23})$$

Calcula el determinante de la matriz m=((2,1,-1,2),(3,2,0,0),(1,2,3,-2),(-1,1,1,1))

```
(%i24) determinant(matrix([2,1,-1,2],[3,2,0,0],[1,2,3,-2],[-1,1,1,1]));
```

$$13 \qquad (\%o_{24})$$

También podemos calcular inversas y traspuestas.

```
(%i25) m:matrix([1,1,1],[2,-1,1],[0,1,1]);
```

$$\begin{pmatrix} 1 & 1 & 1 \\ 2 & -1 & 1 \\ 0 & 1 & 1 \end{pmatrix} \qquad (\%o_{25})$$

```
(%i26) invert(m);
```

$$\begin{pmatrix} 1 & 0 & -1 \\ 1 & -\left(\frac{1}{2}\right) & -\left(\frac{1}{2}\right) \\ -1 & \frac{1}{2} & \frac{3}{2} \end{pmatrix} \qquad (\%o_{26})$$

```
(%i27) transpose(m);
```

$$\begin{pmatrix} 1 & 2 & 0 \\ 1 & -1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \qquad (\%o_{27})$$

Ejercicio

Calcula la inversa de m=((2,3,1),(1,1,1),(2,1,1))

```
(%i28) m:matrix([2,3,1],[1,1,1],[2,1,1]);
```

$$\begin{pmatrix} 2 & 3 & 1 \\ 1 & 1 & 1 \\ 2 & 1 & 1 \end{pmatrix} \qquad (\%o_{28})$$

```
(%i29) determinant(m);
```

$$2 \qquad (\%o_{29})$$

Algunas matrices especiales tienen también comandos para definirlas.

Por ejemplo, la matriz identidad.

```
(%i30) ident(3);
```

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad (\%o_{30})$$

```
(%i31) ident(10);
```

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \qquad (\%o_{31})$$

La extracción de submatrices se realiza con el comando submatrix(filasqueeliminar, nombrematriz, columnasqueeliminar). Veamos algunos ejemplos con la matriz nueva2 definida anteriormente.

```
(%i32) mat:matrix([1,2,2,2],[2,1,-2,2],[0,1,2,1]);
```

$$\begin{pmatrix} 1 & 2 & 2 & 2 \\ 2 & 1 & -2 & 2 \\ 0 & 1 & 2 & 1 \end{pmatrix} \qquad (\%o_{32})$$

```
(%i33) s1:submatrix(2,mat,2,4);
```

$$\begin{pmatrix} 1 & 2 \\ 0 & 2 \end{pmatrix} \qquad (\%o_{33})$$

```
(%i34) s2:submatrix(3,mat);
```

$$\begin{pmatrix} 1 & 2 & 2 & 2 \\ 2 & 1 & -2 & 2 \end{pmatrix} \qquad (\%o_{34})$$

```
(%i35) s3:submatrix(mat,1,4);
```

$$\begin{pmatrix} 2 & 2 \\ 1 & -2 \\ 1 & 2 \end{pmatrix} \qquad (\%o_{35})$$

Ejercicio

Estudia la independencia lineal de los vectores (1,a,1,1),(2,-1,a,1),(-1,1,1,-a) en función del parámetro a.

```
(%i36) m:transpose(matrix([1,a,1,1],[2,-1,a,1],[-1,1,1,-a]));
```

$$\begin{pmatrix} 1 & 2 & -1 \\ a & -1 & 1 \\ 1 & a & 1 \\ 1 & 1 & -a \end{pmatrix} \qquad (\%o_{36})$$

```
(%i37) subm1:submatrix(3,4,m,1);
```

$$\begin{pmatrix} 2 & -1 \\ -1 & 1 \end{pmatrix} \qquad (\%o_{37})$$

```
(%i38) determinant(subm1);
```

$$1 \qquad (\%o_{38})$$

```
(%i39) sub3m1:submatrix(4,m);
```

$$\begin{pmatrix} 1 & 2 & -1 \\ a & -1 & 1 \\ 1 & a & 1 \end{pmatrix} \qquad (\%o_{39})$$

```
(%i40) d1:expand(determinant(sub3m1));
```

$$-a^2 - 3\,a \qquad (\%o_{40})$$

```
(%i41) s1:solve(d1=0,a);
```

$$[a = -3, a = 0] \qquad (\%o_{41})$$

```
(%i42) sub3m2:submatrix(3,m);
```

$$\begin{pmatrix} 1 & 2 & -1 \\ a & -1 & 1 \\ 1 & 1 & -a \end{pmatrix} \qquad (\%o_{42})$$

```
(%i43) d2:expand(determinant(sub3m2));
```

$$2\,a^2 \qquad (\%o_{43})$$

Cuando a=0 el rango es 2, y cuando es distinto es 3

Los comandos addcol y addrow permiten añadir columnas y filas a matrices ya definidas. Observa los ejemplos siguientes.

```
(%i44) kill(all);
```

done $(\%o_{44})$

```
(%i45) m:matrix([1,2,2],[3,2,2]);
```

$$\begin{pmatrix} 1 & 2 & 2 \\ 3 & 2 & 2 \end{pmatrix} \qquad (\%o_{45})$$

```
(%i46) nueva1:addrow(m,[1,2,-1]);
```

$$\begin{pmatrix} 1 & 2 & 2 \\ 3 & 2 & 2 \\ 1 & 2 & -1 \end{pmatrix} \qquad (\%o_{46})$$

```
(%i47) nueva2:addcol(nueva1,[1,2,2]);
```

$$\begin{pmatrix} 1 & 2 & 2 & 1 \\ 3 & 2 & 2 & 2 \\ 1 & 2 & -1 & 2 \end{pmatrix} \qquad (\%o_{47})$$

Para generar matrices grandes a partir de fórmulas, se utiliza el comando genmatrix. Veamos un par de ejemplos.

```
(%i48) coef[i,j]:=i-j;
```

$coef_{i,j} := i - j$ (%o48)

(%i49) `m:genmatrix(coef,3,4);`

$$\begin{pmatrix} 0 & -1 & -2 & -3 \\ 1 & 0 & -1 & -2 \\ 2 & 1 & 0 & -1 \end{pmatrix}$$ (%o49)

En el siguiente ejemplo usamos el condicional para definir diferentes tipos de coeficientes.

(%i50) `co[i,j]:= if i=j then 2 else if i=j+1 then 1 else if i=j-1 then -1 else 0;`

$co_{i,j} := \textbf{if } i = j \textbf{ then } 2 \textbf{ else } (\textbf{if } i = j + 1 \textbf{ then } 1 \textbf{ else } (\textbf{if } i = j - 1 \textbf{ then } -1 \textbf{ else } 0))$ (%o50)

(%i51) `m:genmatrix(co,4,4);`

$$\begin{pmatrix} 2 & -1 & 0 & 0 \\ 1 & 2 & -1 & 0 \\ 0 & 1 & 2 & -1 \\ 0 & 0 & 1 & 2 \end{pmatrix}$$ (%o51)

Ejercicio

Genera una matriz de orden 10 cuyos coeficientes sean 0 en la diagonal, 1 por encima de la diagonal y 3 por debajo de la diagonal. Calcula su determinante y su inversa.

(%i52) `c[i,j]:=if i=j then 0 else if i<j then 1 else if i>j then 3;`

$c_{i,j} := \textbf{if } i = j \textbf{ then } 0 \textbf{ else } (\textbf{if } i < j \textbf{ then } 1 \textbf{ else } (\textbf{if } i > j \textbf{ then } 3))$ (%o52)

(%i53) `m:genmatrix(c,10,10);`

$$\begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 3 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 3 & 3 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 3 & 3 & 3 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 3 & 3 & 3 & 3 & 0 & 1 & 1 & 1 & 1 & 1 \\ 3 & 3 & 3 & 3 & 3 & 0 & 1 & 1 & 1 & 1 \\ 3 & 3 & 3 & 3 & 3 & 3 & 0 & 1 & 1 & 1 \\ 3 & 3 & 3 & 3 & 3 & 3 & 3 & 0 & 1 & 1 \\ 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 0 & 1 \\ 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 0 \end{pmatrix}$$ (%o53)

(%i54) `c[i,j]:=if i=j then (-1)^(i+1) else if i=j+1 then 2*i-3 else if i=1 and j=6 then 3 else 0;`

$c_{i,j} := \textbf{if } i = j \textbf{ then } (-1)^{i+1} \textbf{ else } (\textbf{if } i = j + 1 \textbf{ then } 2\,i - 3 \textbf{ else } (\textbf{if } i = 1 \wedge j = 6 \textbf{ then } 3 \textbf{ else } 0))$ (%o54)

(%i55) `m:genmatrix(c,6,6);`

$$\begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 \\ 3 & 0 & 1 & 1 & 1 & 1 \\ 3 & 3 & 0 & 1 & 1 & 1 \\ 3 & 3 & 3 & 0 & 1 & 1 \\ 3 & 3 & 3 & 3 & 0 & 1 \\ 3 & 3 & 3 & 3 & 3 & 0 \end{pmatrix}$$ (%o55)

(%i56) `c[i,j]:=if i=j then 2^(i-1) else if i>j then -1 else if j=6 then 1 else 0;`

$c_{i,j} := \textbf{if } i = j \textbf{ then } 2^{i-1} \textbf{ else } (\textbf{if } i > j \textbf{ then } -1 \textbf{ else } (\textbf{if } j = 6 \textbf{ then } 1 \textbf{ else } 0))$ (%o56)

Ejercicio

Haz una matriz con una diagonal principal 1 y -1 alternando, arriba de la diagonal principal 2, abajo de la diagonal principal -2 y luego, todo lo demás 0 menos el numero de arriba del todo a la derecha es 3.

```
(%i57) c[i,j]:= if i=j then (-1)^(i+1) else if j=i+1 then 2 else if i=j+1 then -2  else
if (i=1) and (j=6) then 3 else 0;
```

$$c_{i,j} := \textbf{if } i=j \textbf{ then } (-1)^{i+1} \textbf{ else } (\textbf{if } j=i+1 \textbf{ then } 2 \textbf{ else } (\textbf{if } i=j+1 \textbf{ then } -2 \textbf{ else } (\textbf{if } i=1 \wedge j=6 \textbf{ then } 3 \textbf{ else } 0)))) \quad (\%o_{57})$$

Ejercicio

Quiero los numeros pares empezando por el 2 y terminando por el 16 (diagonal principal), los impares (arriba de la diagonal principal), luego todo ceros menos abajo del todo a la izquierda un 10. Calcular su determinante y su inversa.

```
(%i58) c[i,j]:= if i=j then 2*i else if j=i+1 then 2*i-1 else if j>i then 0 else if (i=8)
and (j=1) then 10 else 0;
```

$$c_{i,j} := \textbf{if } i=j \textbf{ then } 2\,i \textbf{ else } (\textbf{if } j=i+1 \textbf{ then } 2\,i-1 \textbf{ else } (\textbf{if } j>i \textbf{ then } 0 \textbf{ else } (\textbf{if } i=8 \wedge j=1 \textbf{ then } 10 \textbf{ else } 0))) \quad (\%o_{58})$$

```
(%i59) m:genmatrix(c,8,8);
```

$$\begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 3 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 3 & 3 & 0 & 1 & 1 & 1 & 1 & 1 \\ 3 & 3 & 3 & 0 & 1 & 1 & 1 & 1 \\ 3 & 3 & 3 & 3 & 0 & 1 & 1 & 1 \\ 3 & 3 & 3 & 3 & 3 & 0 & 1 & 1 \\ 3 & 3 & 3 & 3 & 3 & 3 & 0 & 1 \\ 3 & 3 & 3 & 3 & 3 & 3 & 3 & 0 \end{pmatrix} \quad (\%o_{59})$$

```
(%i60) float(invert(m));
```

$$\begin{pmatrix} -0.333 & 0.2223 & 0.07411 & 0.0247 & 0.008234 & 0.002745 & 9.149\times10^{-4} & 3.05\times10^{-4} \\ 9.149\times10^{-4} & -0.333 & 0.2223 & 0.07411 & 0.0247 & 0.008234 & 0.002745 & 9.149\times10^{-4} \\ 0.002745 & 9.149\times10^{-4} & -0.333 & 0.2223 & 0.07411 & 0.0247 & 0.008234 & 0.002745 \\ 0.008234 & 0.002745 & 9.149\times10^{-4} & -0.333 & 0.2223 & 0.07411 & 0.0247 & 0.008234 \\ 0.0247 & 0.008234 & 0.002745 & 9.149\times10^{-4} & -0.333 & 0.2223 & 0.07411 & 0.0247 \\ 0.07411 & 0.0247 & 0.008234 & 0.002745 & 9.149\times10^{-4} & -0.333 & 0.2223 & 0.07411 \\ 0.2223 & 0.07411 & 0.0247 & 0.008234 & 0.002745 & 9.149\times10^{-4} & -0.333 & 0.2223 \\ 0.667 & 0.2223 & 0.07411 & 0.0247 & 0.008234 & 0.002745 & 9.149\times10^{-4} & -0.333 \end{pmatrix} \quad (\%o_{60})$$

4.2. Ejercicios propuestos

4.2.1. Halla las ecuaciones implícitas de U=<(1,2,1),(3,-1,-1)>. Calcula una base ortogonal de R^3 que tenga dos vectores en U y calcula las coordenadas del vector h=(0,3,2) respecto de la base hallada.

```
(%i1) kill(all);
```

done $(\%o_1)$

```
(%i2) u1: [1,2,1];
```

$[1,2,1]$ $(\%o_2)$

```
(%i3) u2: [3,-1,-1];
```

$[3,-1,-1]$ $(\%o_3)$

```
(%i4) mat: matrix([x,1,3],[y,2,-1],[z,1,-1]);
```

$$\begin{pmatrix} x & 1 & 3 \\ y & 2 & -1 \\ z & 1 & -1 \end{pmatrix} \qquad (\%o_4)$$

```
(%i5) ec: expand(determinant(mat)) = 0;
```

$$-(7z)+4y-x=0 \qquad (\%o_5)$$

```
(%i6) v1: u1;
```

$$[1,2,1] \qquad (\%o_6)$$

```
(%i7) v2: u2;
```

$$[3,-1,-1] \qquad (\%o_7)$$

```
(%i8) v3: [-1, 4, -7];
```

$$[-1,4,-7] \qquad (\%o_8)$$

```
(%i9) v3.v1;
```

$$0 \qquad (\%o_9)$$

```
(%i10) v3.v2;
```

$$0 \qquad (\%o_{10})$$

```
(%i11) h: [0,3,2];
```

$$[0,3,2] \qquad (\%o_{11})$$

```
(%i12) c1: (h.v1) / (v1.v1);
```

$$\frac{4}{3} \qquad (\%o_{12})$$

```
(%i13) c2: (h.v2) / (v2.v2);
```

$$-\left(\frac{5}{11}\right) \qquad (\%o_{13})$$

```
(%i14) c3: (h.v3) / (v3.v3);
```

$$-\left(\frac{1}{33}\right) \qquad (\%o_{14})$$

```
(%i15) coordenadas: [c1, c2, c3];
```

$$\left[\frac{4}{3},-\left(\frac{5}{11}\right),-\left(\frac{1}{33}\right)\right] \qquad (\%o_{15})$$

4.2.2. Calcula la proyección ortogonal sobre la recta R: x+y-z=0, 3x-2y-z=0. Calcula también la simetría ortogonal respecto de R. De los vectores r1=(1,2,1) y r2=(0,1,2), ¿cuál está más cerca de la recta R?

```
(%i16) kill(all);
```

done (%o16)

```
(%i17) n1: [1, 1, -1];
```

$$[1,1,-1] \qquad (\%o_{17})$$

```
(%i18) n2: [3, -2, -1];
```

$$[3,-2,-1] \qquad (\%o_{18})$$

```
(%i19) vr: [    n1[2]*n2[3] - n1[3]*n2[2],    n1[3]*n2[1] - n1[1]*n2[3],    n1[1]*n2[2]
- n1[2]*n2[1]  ];
```

$$[-3,-2,-5] \qquad (\%o_{19})$$

```
(%i20) Proyeccion(v, u):= (v.u)/(u.u) * u;
```

$$Proyeccion\,(v,u) := \frac{v\cdot u}{u\cdot u}\,u \qquad (\%o_{20})$$

```
(%i21) Simetria(v, u):= 2 * Proyeccion(v, u) - v;
```

$$Simetria\,(v,u) := 2\,Proyeccion\,(v,u) - v \qquad (\%o_{21})$$

```
(%i22) r1: [1, 2, 1];
```

$$[1,2,1] \qquad (\%o_{22})$$

```
(%i23) r2: [0, 1, 2];
```

$$[0,1,2] \qquad (\%o_{23})$$

```
(%i24) P_r1: Proyeccion(r1, vr);
```

$$\left[\frac{18}{19},\frac{12}{19},\frac{30}{19}\right] \qquad (\%o_{24})$$

```
(%i25) P_r2: Proyeccion(r2, vr);
```

$$\left[\frac{18}{19},\frac{12}{19},\frac{30}{19}\right] \qquad (\%o_{25})$$

```
(%i26) S_r1: Simetria(r1, vr);
```

$$\left[\frac{17}{19},-\left(\frac{14}{19}\right),\frac{41}{19}\right] \qquad (\%o_{26})$$

```
(%i27) S_r2: Simetria(r2, vr);
```

$$\left[\frac{36}{19},\frac{5}{19},\frac{22}{19}\right] \qquad (\%o_{27})$$

```
(%i28) norma(v):= sqrt(v.v);
```

$$norma\,(v) := \sqrt{v\cdot v} \qquad (\%o_{28})$$

```
(%i29) dist_r1: float(norma(r1 - P_r1));
```

$$1.4867838833500562 \qquad (\%o_{29})$$

```
(%i30) dist_r2: float(norma(r2 - P_r2));
```

$$1.1002392084403616 \qquad (\%o_{30})$$

r2 está más cerca

4.2.3. Considera la aplicación lineal f(x,y,z)=(x+3y, y+z, x+y-z,x+3z) de R^3 en R^4. Define la matriz que representa a f respecto de las bases usuales. Estudia los valores de a para los cuales el vector h=(1,a,2a,0) está en la imagen de f.

```
(%i31) M: matrix(    [1, 3, 0],    [0, 1, 1],    [1, 1, -1],    [1, 0, 3] );
```

$$\begin{pmatrix}1 & 3 & 0\\ 0 & 1 & 1\\ 1 & 1 & -1\\ 1 & 0 & 3\end{pmatrix} \qquad (\%o_{31})$$

```
(%i32) h: [1, a, 2*a, 0];
```

$$[1,a,2\,a,0] \qquad (\%o_{32})$$

```
(%i33) M_ampliada: addcol(M, h);
```

$$\begin{pmatrix} 1 & 3 & 0 & 1 \\ 0 & 1 & 1 & a \\ 1 & 1 & -1 & 2\,a \\ 1 & 0 & 3 & 0 \end{pmatrix} \tag{\%o33}$$

(%i34)
```
det: determinant(M_ampliada);
```

$$5 - 21\,a \tag{\%o34}$$

(%i35)
```
sol: solve(det=0, a);
```

$$\left[a = \frac{5}{21}\right] \tag{\%o35}$$

CAPÍTULO 5

Práctica 5: Diagonalización y factorizaciones

Resumen de la práctica

Práctica final del curso: cálculo de **valores y vectores propios**, estudio de la **diagonalización** de matrices, aplicación al cálculo de potencias y procesos estacionarios, y dos factorizaciones fundamentales del álgebra lineal numérica: **LU** y **QR** (con su uso para aproximar valores propios).

5.1. Diagonalización

Para calcular los valores propios de una matriz disponemos del comando eigenvalues.

```
(%i1) m:matrix([1,1,1],[2,1,0],[-3,3,3]);
```

$$\begin{pmatrix} 1 & 1 & 1 \\ 2 & 1 & 0 \\ -3 & 3 & 3 \end{pmatrix} \qquad (\%o_1)$$

```
(%i2) eigenvalues(m);
```

$$[[1-i, i+1, 3], [1, 1, 1]] \qquad (\%o_2)$$

En la primera sublista aparecen los valores propios (reales y complejos). En la segunda sublista se muestran las multiplicidades algebraicas.

El único valor propio real de la matriz es 3. No es diagonalizable por tanto.

Con el comando eigenvectors, podemos calcular también los vectores propios.

```
(%i3) eigenvectors(m);
```

$$[[[1-i, i+1, 3], [1, 1, 1]], [[[1, 2\,i, -(3\,i)]], [[1, -(2\,i), 3\,i]], [[1, 1, 1]]]] \qquad (\%o_3)$$

Si queremos calcular el polinomio característico, podemos proceder así:

```
(%i4) p:charpoly(m,x);
```

$$(1-x)^2\,(3-x) - 2\,(3-x) + 3\,(1-x) + 6 \qquad (\%o_4)$$

```
(%i5) expand(p);
```

$$-x^3 + 5\,x^2 - 8\,x + 6 \qquad (\%o_5)$$

Ejercicio

Estudia si son diagonalizables las matrices m=((1,2,1),(2,0,2),(1,2,1)), n=((2,-2,1),(2,1,-2),(-1,-1,3)) o=((1,1,1),(0,1,2),(0,0,2))

```
(%i6) m:matrix([1,2,1],[2,0,2],[1,2,1]);
```

$$\begin{pmatrix}1 & 2 & 1\\ 2 & 0 & 2\\ 1 & 2 & 1\end{pmatrix} \qquad (\%o_6)$$

```
(%i7) n:matrix([2,-2,1],[2,1,-2],[-1,-1,3]);
```

$$\begin{pmatrix}2 & -2 & 1\\ 2 & 1 & -2\\ -1 & -1 & 3\end{pmatrix} \qquad (\%o_7)$$

```
(%i8) eigenvectors(n);
```

$$\left[\left[\left[-\left(\frac{\sqrt{11}\,i-5}{2}\right),\frac{\sqrt{11}\,i+5}{2},1\right],[1,1,1]\right],\left[\left[\left[1,\frac{2\,i}{\sqrt{11}},-\left(\frac{3\sqrt{11}\,i-11}{22}\right)\right]\right],\left[\left[1,-\left(\frac{2\,i}{\sqrt{11}}\right),\frac{3\sqrt{11}\,i+11}{22}\right]\right],[[1,1,1]]\right]\right] \qquad (\%o_8)$$

```
(%i9) o:matrix([1,1,1],[0,1,2],[0,0,2]);
```

$$\begin{pmatrix}1 & 1 & 1\\ 0 & 1 & 2\\ 0 & 0 & 2\end{pmatrix} \qquad (\%o_9)$$

```
(%i10) eigenvectors(o);
```

$$\left[[[1,2],[2,1]],\left[[[1,0,0]],\left[\left[1,\frac{2}{3},\frac{1}{3}\right]\right]\right]\right] \qquad (\%o_{10})$$

Estudia si es diangonalizable la matriz M dada por ((1,2,3),(2,5,4),(3,4,-6))

```
(%i11) m:matrix([1,2,3],[2,5,4],[3,4,-6]);
```

$$\begin{pmatrix}1 & 2 & 3\\ 2 & 5 & 4\\ 3 & 4 & -6\end{pmatrix} \qquad (\%o_{11})$$

```
(%i12) vp:float(rectform(eigenvectors(m)));
```

$$[[[8.882\times10^{-16}\,i+0.3172,-\left(4.441\times10^{-16}\,i\right)-7.9,7.582-4.441\times10^{-16}\,i],[1.0,1.0,1.0]],[[[1.0,2.479\times10^{-16}\,i-0.5404,0.1327]],[[1.0,0.9642,5.288\times10^{-16}\,i-3.609]],[[1.0,2.053-2.115\times10^{-16}\,i,0.8255-4.23\times10^{-16}\,i]]]] \qquad (\%o_{12})$$

```
(%i13) realpart(vp);
```

$$[[[0.3172,-7.9,7.582],[1.0,1.0,1.0]],[[[1.0,-0.5404,0.1327]],[[1.0,0.9642,-3.609]],[[1.0,2.053,0.8255]]]] \qquad (\%o_{13})$$

Estudia m=((1,2,6),(0,-7,2),(-2,0,4))

```
(%i14) m:matrix([1,2,6],[0,-7,2],[-2,0,4]);
```

$$\begin{pmatrix}1 & 2 & 6\\ 0 & -7 & 2\\ -2 & 0 & 4\end{pmatrix} \qquad (\%o_{14})$$

Veamos ahora un ejemplo en el que además extraemos la matriz de cambio de base.

Sea

```
(%i15) kill(all);
```

done $\qquad (\%o_{15})$

```
(%i16) A:matrix([1,1,1],[1,4,-1],[1,4,1]);
```

$$\begin{pmatrix}1 & 1 & 1\\ 1 & 4 & -1\\ 1 & 4 & 1\end{pmatrix} \qquad (\%o_{16})$$

```
(%i17) vp:eigenvectors(A);
```

$$\left[[[1,2,3],[1,1,1]],\left[\left[\left[1,-\left(\frac{1}{4}\right),\frac{1}{4}\right]\right],[[1,0,1]],\left[\left[1,\frac{1}{2},\frac{3}{2}\right]\right]\right]\right] \qquad (\%o_{17})$$

Los vectores de la base son

```
(%i18) v1:vp[2][1][1];
```

$$\left[1,-\left(\frac{1}{4}\right),\frac{1}{4}\right] \tag{\%o18}$$

```
(%i19) v2:vp[2][2][1];
```

$$[1,0,1] \tag{\%o19}$$

```
(%i20) v3:vp[2][3][1];
```

$$\left[1,\frac{1}{2},\frac{3}{2}\right] \tag{\%o20}$$

La matriz de cambio de base es

```
(%i21) P:transpose(matrix(v1,v2,v3));
```

$$\begin{pmatrix} 1 & 1 & 1 \\ -\left(\frac{1}{4}\right) & 0 & \frac{1}{2} \\ \frac{1}{4} & 1 & \frac{3}{2} \end{pmatrix} \tag{\%o21}$$

Comprobamos la diagonalización

```
(%i22) D:invert(P).A.P;
```

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{pmatrix} \tag{\%o22}$$

Utilicemos esto para calcular A^8. Lo deducimos de que A=PDP^(-1)

```
(%i23) D^8;
```

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 256 & 0 \\ 0 & 0 & 6561 \end{pmatrix} \tag{\%o23}$$

```
(%i24) A8:P.D^(8).invert(P);
```

$$\begin{pmatrix} 6051 & 18405 & -5795 \\ 3280 & 9841 & -3280 \\ 9330 & 28245 & -9074 \end{pmatrix} \tag{\%o24}$$

Ejercicio

Dos empresas energéticas suministran luz a una población fija. La empresa A observa que, año a año, conserva al 80 % de sus clientes, mientras que el 20 % se cambia a la empresa B. Por otro lado, la empresa B conserva al 60 % de sus clientes, mientras que el resto se cambia a la empresa A. Si inicialmente ambas empresas se reparten a la población al 50 %, halla la distribución de los clientes al cabo de 10 años. ¿Qué ocurre cuando pasa mucho tiempo?

```
(%i25) M: matrix([0.8, 0.4], [0.2, 0.6]);
```

$$\begin{pmatrix} 0.8 & 0.4 \\ 0.2 & 0.6 \end{pmatrix} \tag{\%o25}$$

```
(%i26) v0: matrix([0.5], [0.5]);
```

$$\begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix} \tag{\%o26}$$

```
(%i27) v10: float((M^^10) . v0);
```

$$\begin{pmatrix} 0.6666491904000003 \\ 0.3333508096000001 \end{pmatrix} \quad (\%o_{27})$$

```
(%i28) print("Empresa A (aprox): ", v10[1,1]);
```

```
Empresa A (aprox): 0.6666491904000003
```

0.6666491904000003 (%o28)

```
(%i29) print("Empresa B (aprox): ", v10[2,1]);
```

```
Empresa B (aprox): 0.3333508096000001
```

0.3333508096000001 (%o29)

```
(%i30) [vals, vects]: eigenvectors(M);
```

$$\left[\left[\left[-\left(\frac{9\sqrt{360576170731585314253279314899}-12610078956637389}{180143985094819 84}\right)\cdot\frac{9\sqrt{360576170731585314253279314899}+12610078956637389}{180143985094819 84}\right],[1,1]\right],\left[\left[\left[1,-\left(\frac{9\sqrt{360576170731585314253279314899}+180143985094819 9}{7205759403792794}\right)\right]\right],\left[\left[1,\frac{9\sqrt{360576170731585314253279314899}-180143985094819 9}{7205759403792794}\right]\right]\right] \quad (\%o_{30})$$

```
(%i31) v_final: float((M^^50) . v0);
```

$$\begin{pmatrix} 0.6666666666666679 \\ 0.3333333333333337 \end{pmatrix} \quad (\%o_{31})$$

```
(%i32) print("-----------------");
```

```
-----------------
```

(%o32)

```
(%i33) print("A largo plazo (Estado Estacionario):");
```

```
A largo plazo (Estado Estacionario):
```

A largo plazo (Estado Estacionario): (%o33)

```
(%i34) print("Empresa A:", v_final[1,1]);
```

```
Empresa A: 0.6666666666666679
```

0.6666666666666679 (%o34)

```
(%i35) print("Empresa B:", v_final[2,1]);
```

```
Empresa B: 0.3333333333333337
```

0.3333333333333337 (%o35)

5.2. Factorización LU

La factorización LU de una matriz regular se realiza con los siguientes comandos:

```
(%i1) A:matrix([1,2,6],[2,3,0],[1,0,-3]);
```

$$\begin{pmatrix} 1 & 2 & 6 \\ 2 & 3 & 0 \\ 1 & 0 & -3 \end{pmatrix} \quad (\%o_1)$$

```
(%i2) lu:lu_factor(A);
```

$$\left[\begin{pmatrix} 1 & 2 & 6 \\ 2 & -1 & -12 \\ 1 & 2 & 15 \end{pmatrix}, [1,2,3], \mathit{generalring}\right] \quad (\%o_2)$$

Esta es una presentación compacta de la factorización. Para desglosarla, se procede como sigue:

```
(%i3) [p,l,u]:get_lu_factors(lu);
```

$$\left[\begin{pmatrix}1&0&0\\0&1&0\\0&0&1\end{pmatrix},\begin{pmatrix}1&0&0\\2&1&0\\1&2&1\end{pmatrix},\begin{pmatrix}1&2&6\\0&-1&-12\\0&0&15\end{pmatrix}\right] \tag{\%o3}$$

Comprobemos que es efectivamente una factorización de la matriz A.

```
(%i4) l.u;
```

$$\begin{pmatrix}1&2&6\\2&3&0\\1&0&-3\end{pmatrix} \tag{\%o4}$$

Además, la resolución de sistemas lineales basándose en esta factorización se lleva a cabo de la siguiente manera.

```
(%i5) b:matrix([1],[4],[28]);
```

$$\begin{pmatrix}1\\4\\28\end{pmatrix} \tag{\%o5}$$

```
(%i6) sol:lu_backsub(lu,b);
```

$$\begin{pmatrix}\frac{163}{5}\\-\left(\frac{102}{5}\right)\\\frac{23}{15}\end{pmatrix} \tag{\%o6}$$

Si b es una matriz de varias columnas, la solución te da las columnas de las soluciones.

```
(%i7) b:matrix([1,1],[1,1], [1,30]);
```

$$\begin{pmatrix}1&1\\1&1\\1&30\end{pmatrix} \tag{\%o7}$$

```
(%i8) sol2:lu_backsub(lu,b);
```

$$\begin{pmatrix}\frac{7}{5}&\frac{181}{5}\\-\left(\frac{3}{5}\right)&-\left(\frac{119}{5}\right)\\\frac{2}{15}&\frac{31}{15}\end{pmatrix} \tag{\%o8}$$

Vamos a resolver un sistema de orden 20

```
(%i9) kill(all);
```

done (%o9)

```
(%i10) a[i,j]:= if i=j then 2 else if abs(i-j)=1 then 1 else 0;
```

$a_{i,j} := \textbf{if } i=j \textbf{ then } 2 \textbf{ else } (\textbf{if } |i-j|=1 \textbf{ then } 1 \textbf{ else } 0)$ (%o10)

```
(%i11) mat:genmatrix(a,20);
```

$$\begin{pmatrix}
2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2
\end{pmatrix} \quad (\%o_{11})$$

(%i12) `des:lu_factor(mat);`

$$\left[\begin{pmatrix}
2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{2} & \frac{3}{2} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{2}{3} & \frac{4}{3} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \frac{3}{4} & \frac{5}{4} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \frac{4}{5} & \frac{6}{5} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \frac{5}{6} & \frac{7}{6} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \frac{6}{7} & \frac{8}{7} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \frac{7}{8} & \frac{9}{8} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{8}{9} & \frac{10}{9} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{9}{10} & \frac{11}{10} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{10}{11} & \frac{12}{11} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{11}{12} & \frac{13}{12} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{12}{13} & \frac{14}{13} & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{13}{14} & \frac{15}{14} & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{14}{15} & \frac{16}{15} & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{15}{16} & \frac{17}{16} & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{16}{17} & \frac{18}{17} & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{17}{18} & \frac{19}{18} & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{18}{19} & \frac{20}{19} & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{19}{20} & \frac{21}{20}
\end{pmatrix}, [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20], generalring\right] \quad (\%o_{12})$$

(%i13) `[p,l,u]:get_lu_factors(des);`

$$\left[\begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix}, \begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{2} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{2}{3} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \frac{3}{4} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \frac{4}{5} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \frac{5}{6} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \frac{6}{7} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \frac{7}{8} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{8}{9} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{9}{10} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{10}{11} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{11}{12} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{12}{13} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{13}{14} & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{14}{15} & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{15}{16} & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{16}{17} & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{17}{18} & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{18}{19} & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{19}{20} & 1
\end{pmatrix}, \begin{pmatrix}
2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{3}{2} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \frac{4}{3} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \frac{5}{4} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \frac{6}{5} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \frac{7}{6} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \frac{8}{7} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{9}{8} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{10}{9} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{11}{10} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{12}{11} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{13}{12} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{14}{13} & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{15}{14} & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{16}{15} & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{17}{16} & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{18}{17} & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{19}{18} & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{20}{19} & 1
\end{pmatrix}\right] \quad (\%o_{13})$$

(%i14) `b[i,j]:=if i<8 then 0 else if (i>=8) and (i<=15 ) then (-1)^i else 2;`

$$b_{i,j} := \textbf{if } i < 8 \textbf{ then } 0 \textbf{ else } \left(\textbf{if } i \geq 8 \wedge i \leq 15 \textbf{ then } (-1)^i \textbf{ else } 2\right) \quad (\%o_{14})$$

(%i15) `indep:genmatrix(b,20,1);`

$$\begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & -1 & 1 & -1 & 1 \\
-1 & 1 & -1 & 2 & 2 & 2 \\
2 & 2 & & & &
\end{pmatrix}^{\mathsf{T}} \quad (\%o_{15})$$

(%i16) `sol:lu_backsub(des,indep);`

$$\begin{pmatrix} -\left(\frac{82}{21}\right) \\ \frac{164}{21} \\ -\left(\frac{82}{7}\right) \\ \frac{328}{21} \\ -\left(\frac{410}{21}\right) \\ \frac{164}{7} \\ -\left(\frac{82}{3}\right) \\ \frac{656}{21} \\ -\left(\frac{239}{7}\right) \\ \frac{757}{21} \\ -\left(\frac{776}{21}\right) \\ \frac{258}{7} \\ -\left(\frac{751}{21}\right) \\ \frac{101}{3} \\ -\left(\frac{214}{7}\right) \\ \frac{556}{21} \\ -\left(\frac{428}{21}\right) \\ \frac{114}{7} \\ -\left(\frac{214}{21}\right) \\ \frac{128}{21} \end{pmatrix} \qquad (\%o_{16})$$

```
(%i17) float(sol);
```

$$\begin{pmatrix} -3.905 \\ 7.81 \\ -11.71 \\ 15.62 \\ -19.52 \\ 23.43 \\ -27.33 \\ 31.24 \\ -34.14 \\ 36.05 \\ -36.95 \\ 36.86 \\ -35.76 \\ 33.67 \\ -30.57 \\ 26.48 \\ -20.38 \\ 16.29 \\ -10.19 \\ 6.095 \end{pmatrix} \qquad (\%o_{17})$$

```
(%i18) transpose(float(sol));
```

$$\begin{pmatrix} -3.905 & 7.81 & -11.71 & 15.62 & -19.52 & 23.43 \\ -27.33 & 31.24 & -34.14 & 36.05 & -36.95 & 36.86 \\ -35.76 & 33.67 & -30.57 & 26.48 & -20.38 & 16.29 \\ -10.19 & 6.095 & & & & \end{pmatrix} \qquad (\%o_{18})$$

Considera la matriz A de orden 10 que tenga los primeros 10 numeros impares en la diagonal.

1 todos los coeficientes por encima de la diagonal, -1 en el coeficiente (10,1), y cero el resto. Considera B los terminos independientes formados por cuadrados de los primeros numeros naturales.. Resuelve el sistema Ax=B usando la descomposixion LU

```
(%i19) a[i,j]:= if i=j then 2*i else if j>i then 1 else if i=10 and j=1 then -1 else 0;
```

$$a_{i,j} := \textbf{if } i = j \textbf{ then } 2\,i \textbf{ else } (\textbf{if } j > i \textbf{ then } 1 \textbf{ else } (\textbf{if } i = 10 \wedge j = 1 \textbf{ then } -1 \textbf{ else } 0))$$ (%o19)

```
(%i20) A:genmatrix(a,10);
```

$$\begin{pmatrix} 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 2 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 \end{pmatrix}$$ (%o20)

```
(%i21) b[i,j]:= i^2;
```

$$b_{i,j} := i^2$$ (%o21)

```
(%i22) B:genmatrix(b,10,1);
```

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 \end{pmatrix}^{\mathsf{T}}$$ (%o22)

5.3. Factorización QR

Requiere cargar un paquete, como se indica ahora:

```
(%i1) load("lapack");
```

```
      fallado al cargar
      /opt/homebrew/Cellar/maxima/5.49.0_4/share/maxima/5.49.0/share/lapack/load-lapack.lisp
```

! Error: *fallado al cargar /opt/homebrew/Cellar/maxima/5.49.0_4/share/maxima/5.49.0/share/lapack/load-lapack.lisp* (%o1)

Definamos la matriz

```
(%i2) A:matrix([1,2,0],[2,1,4],[0,-2,4]);
```

$$\begin{pmatrix} 1 & 2 & 0 \\ 2 & 1 & 4 \\ 0 & -2 & 4 \end{pmatrix}$$ (%o2)

El comando que se usa para factorizar es

```
(%i3) [q, r] : dgeqrf (A);
```

$$\left[dgeqrf\left(\begin{pmatrix} 1 & 2 & 0 \\ 2 & 1 & 4 \\ 0 & -2 & 4 \end{pmatrix}\right), dgeqrf\left(\begin{pmatrix} 1 & 2 & 0 \\ 2 & 1 & 4 \\ 0 & -2 & 4 \end{pmatrix}\right) \right]$$ (%o3)

Comprobemos que q es ortogonal

```
(%i4) transpose(q).q;
```

$$transpose\left(dgeqrf\left(\begin{pmatrix} 1 & 2 & 0 \\ 2 & 1 & 4 \\ 0 & -2 & 4 \end{pmatrix}\right)\right) \cdot dgeqrf\left(\begin{pmatrix} 1 & 2 & 0 \\ 2 & 1 & 4 \\ 0 & -2 & 4 \end{pmatrix}\right)$$ (%o4)

Veamos que la factorización es cierta.

```
(%i5) q.r;
```

$$dgeqrf\left(\begin{pmatrix}1 & 2 & 0\\ 2 & 1 & 4\\ 0 & -2 & 4\end{pmatrix}\right)^{\langle 2\rangle} \qquad (\%o_5)$$

Factoricemos una matriz grande.

```
(%i6) m[i,j]:=if i=j then 2 else if i=j+1 then (-1)^j else if j=i+1 then 1 else 0;
```

$$m_{i,j} := \textbf{if } i=j \textbf{ then } 2 \textbf{ else } \left(\textbf{if } i=j+1 \textbf{ then } (-1)^j \textbf{ else } (\textbf{if } j=i+1 \textbf{ then } 1 \textbf{ else } 0)\right) \qquad (\%o_6)$$

```
(%i7) mat:genmatrix(m,20);
```

$$\begin{pmatrix}
2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
-1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & -1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & -1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & 1 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 1 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & 1 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 1 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & 1 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 1\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2
\end{pmatrix} \qquad (\%o_7)$$

```
(%i8) [q,r]:dgeqrf(mat);
```

$$\left[dgeqrf\left(\begin{pmatrix}
2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
-1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & -1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & -1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & 1 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 1 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & 1 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 1 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & 1 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 1\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2
\end{pmatrix}\right), dgeqrf\left(\begin{pmatrix}
2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
-1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & -1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & -1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & 1 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 1 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & 1 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 1 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & 1 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 1\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2
\end{pmatrix}\right)\right] \qquad (\%o_8)$$

```
(%i9) q;
```

$$dgeqrf\left(\begin{pmatrix}
2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
-1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & -1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & -1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & 1 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 1 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & 1 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 1 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & 1 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 1\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2
\end{pmatrix}\right) \quad (\%o_9)$$

(%i10) `r;`

$$dgeqrf\left(\begin{pmatrix}
2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
-1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & -1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & -1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & 1 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 1 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & 1 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 1 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & 1 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 1\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2
\end{pmatrix}\right) \quad (\%o_{10})$$

Vamos a utilizar la factorización QR para la aproximación de valores propios. Vamos a hacerlo para una matriz simétrica no demasiado grande.

(%i11) `kill(all);`

done (%o11)

(%i12) `m[i,j]:= if i=j then 2 else if abs(i-j)=1 then 1 else 0;`

$m_{i,j} := \textbf{if } i = j \textbf{ then } 2 \textbf{ else } (\textbf{if } |i-j| = 1 \textbf{ then } 1 \textbf{ else } 0)$ (%o12)

(%i13) `A:genmatrix(m,3);`

$$\begin{pmatrix} 2 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 2 \end{pmatrix} \quad (\%o_{13})$$

Programemos el método.

```
(%i14) B:A;
```

$$\begin{pmatrix} 2 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 2 \end{pmatrix} \quad (\%o_{14})$$

```
(%i15) for i:1 thru 30 do ([q,r]:dgeqrf(B),B:r.q);
```

done (%o15)

```
(%i16) display(B);
```

```
B =
dgeqrf(dgeqrf(dgeqrf(dgeqrf(dgeqrf(dgeqrf(dgeqrf(dgeqrf(dgeqrf(dgeqrf(dgeqrf(dgeqrf(dge
dgeqrf(  dgeqrf(  dgeqrf(  dgeqrf(  dgeqrf(  dgeqrf(  dgeqrf(  dgeqrf(
dgeqrf(  dgeqrf(  dgeqrf(
dgeqrf(dgeqrf(dgeqrf(dgeqrf(dgeqrf(matrix([2,1,0],[1,2,1],[0,1,2]))^^2)^^2)^^2)^^2)^^2)
^^2)  ^^2)  ^^2)  ^^2)  ^^2)  ^^2)  ^^2)  ^^2)  ^^2)  ^^2)  ^^2)  ^^2)
^^2)  ^^2)  ^^2)  ^^2)  ^^2)  ^^2)  ^^2)  ^^2)  ^^2)  ^^2)  ^^2
```

done (%o16)

```
(%i17) float(eigenvectors(A));
```

$$[[[0.5858, 3.414, 2.0], [1.0, 1.0, 1.0]], [[[1.0, -1.414, 1.0]], [[1.0, 1.414, 1.0]], [[1.0, 0.0, -1.0]]]] \quad (\%o_{17})$$

Ejercicio

Considera la matriz A de orden 8 que tiene -1 y 1 altermos en la diagonal, 1 justamwewnte por encima y por debajo de la diagonal, y en las esquinas superior derecha e inferior izquierda igual a 5. Calcula sus valores propios con el metodo QR.

```
(%i18) a[i,j]:= if i=j then (-1)^(i+1) else if abs(i-j)=1 then 1 else if i=1 and j=8 then
5 else if i=8 and j=1 then 5 else 0;
```

$$a_{i,j} := \textbf{if } i = j \textbf{ then } (-1)^{i+1} \textbf{ else } (\textbf{if } |i-j| = 1 \textbf{ then } 1 \textbf{ else } (\textbf{if } i = 1 \wedge j = 8 \textbf{ then } 5 \textbf{ else } (\textbf{if } i = 8 \wedge j = 1 \textbf{ then } 5 \textbf{ else } 0))) \quad (\%o_{18})$$

```
(%i19) A:genmatrix(a,8);
```

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 5 \\ 1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 5 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{pmatrix} \quad (\%o_{19})$$

Ejercicio

Cambia la matriz anterior de manera que en la diagonal aparezcan los primeros 8 numeros pares.

```
(%i20) a[i,j]:= if i=j then 2*i else if abs(i-j)=1 then 1 else if i=1 and j=8 then 5 else
if i=8 and j=1 then 5 else 0;
```

$$a_{i,j} := \textbf{if } i = j \textbf{ then } 2\,i \textbf{ else } (\textbf{if } |i-j| = 1 \textbf{ then } 1 \textbf{ else } (\textbf{if } i = 1 \wedge j = 8 \textbf{ then } 5 \textbf{ else } (\textbf{if } i = 8 \wedge j = 1 \textbf{ then } 5 \textbf{ else } 0))) \quad (\%o_{20})$$

```
(%i21) B:A;
```

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 5 \\ 1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 5 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{pmatrix} \qquad (\%o_{21})$$

```
(%i22) for i:1 thru 50 do ([q,r]:dgeqrf(B),B:r.q);
```

done (%o22)

```
(%i23) display(B);
```

```
                                B =
dgeqrf(dgeqrf(dgeqrf(dgeqrf(dgeqrf(dgeqrf(dgeqrf(dgeqrf(dgeqrf(dgeqrf(dgeqrf(dgeqrf(dgeq
dgeqrf(  dgeqrf(  dgeqrf(  dgeqrf(  dgeqrf(  dgeqrf(  dgeqrf(  dgeqrf(
dgeqrf(  dgeqrf(  dgeqrf(  dgeqrf(  dgeqrf(  dgeqrf(  dgeqrf(  dgeqrf(
dgeqrf(  dgeqrf(  dgeqrf(  dgeqrf(  dgeqrf(  dgeqrf(  dgeqrf(  dgeqrf(
dgeqrf(  dgeqrf(  dgeqrf(  dgeqrf(  dgeqrf(  dgeqrf(  dgeqrf(  dgeqrf(
dgeqrf(  dgeqrf(  dgeqrf(  dgeqrf(
matrix([1,1,0,0,0,0,0,5],[1,-1,1,0,0,0,0,0],[0,1,1,1,0,0,0,0],
[0,0,1,-1,1,0,0,0],[0,0,0,1,1,1,0,0],[0,0,0,0,1,-1,1,0],
[0,0,0,0,0,1,1,1],[5,0,0,0,0,0,1,-1]))  ^^2)  ^^2)  ^^2)  ^^2)  ^^2)  ^^2)
^^2)  ^^2)  ^^2)  ^^2)  ^^2)  ^^2)  ^^2)  ^^2)  ^^2)  ^^2)  ^^2)  ^^2)
^^2)  ^^2)  ^^2)  ^^2)  ^^2)  ^^2)  ^^2)  ^^2)  ^^2)  ^^2)  ^^2)  ^^2)
^^2)  ^^2)  ^^2)  ^^2)  ^^2)  ^^2)  ^^2)  ^^2)  ^^2)  ^^2)  ^^2)  ^^2)
^^2)  ^^2)  ^^2)  ^^2)  ^^2
```

done (%o23)

FIN DEL LIBRO

Compilado íntegramente con LaTeX y Maxima

David Conesa Pagán ▪ UPCT ▪ Matemáticas I